

Trabajo Fin de Carrera

Grado en Ingeniería de las Tecnologías de Telecomunicación

Análisis e identificación de potenciales evocados en el electroencefalograma

Autor: Elena María Mosquera Guerrero

Tutor: Rubén Martín Clemente

Sergio Antonio Cruces Álvarez

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019



Trabajo Fin de Carrera
Grado en Ingeniería de las Tecnologías de Telecomunicación

Análisis e identificación de potenciales evocados en el electroencefalograma

Autor:

Elena María Mosquera Guerrero

Tutor:

Rubén Martín Clemente

Profesor titular

Sergio Antonio Cruces Álvarez

Catedrático

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019

Proyecto Fin de Carrera: Análisis e identificación de potenciales evocados en el electroencefalograma

Autor: Elena María Mosquera Guerrero

Tutor: Rubén Martín Clemente
Sergio Antonio Álvarez Cruces

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

Agradecimientos

Aunque esté al principio de la memoria, he decidido escribir este apartado en el último momento. Estar redactando estas palabras tiene para mí muchas connotaciones y emoción: estoy a unos pocos pasos de terminar esta carrera que tanto esfuerzo ha requerido.

Siquiera sé cómo comenzar a escribir lo agradecida que estoy de todo lo que me llevo conmigo a un siguiente paso en mi vida, quizás un poco incierto. Quiero hablarle al ente abstracto de la carrera, por haberme resultado un reto intelectual y personal, por haberme hecho crecer, madurar, aceptar, trabajar y otras tantas habilidades que ni sé explicar.

Me dirijo también a todos los que me han acompañado durante este arduo camino, estando conmigo en las mejores, y en las peores. A mis profesores, hablando como una alumna más, porque todos ellos han contribuido a que esté donde estoy hoy, forman parte de mi corazón y mi persona. A mi grupo de compañeras y amigas, porque sin ellas... Ni siquiera sé dónde estaría sin ellas. A mis compañeros, por ayudarme a trabajar en equipo, a romper barreras, a ser mejor persona, ayudante, líder, técnica.

Y, por supuesto, a mi familia. Mis padres, mi hermano. No sé cómo expresar este profundo sentimiento que tengo al pensar en cómo me habéis ayudado, apoyado, animado, en cómo hemos celebrado y cómo habéis llorado conmigo en los momentos difíciles.

A todas y cada una de estas personas, por ayudarme a construir las bonitas alas que tengo para explorar todo este mundo nuevo que me espera.

Elena María Mosquera Guerrero

Sevilla, 2019

Resumen

Las interfaces cerebro-computadora (BCI) sugieren un gran avance para la medicina. Entre sus diferentes usos, está la comunicación no muscular para personas discapacitadas. Uno de estas realizaciones se hace mediante la onda P300 generada mediante la matriz descrita por Farwell y Donchin en 1998, estudiando electroencefalogramas, tanto en línea como grabaciones.

En este proyecto, se pretende estudiar algoritmos y métodos para la mejora de la predicción de caracteres pensados por un sujeto, de manera que se investigue la optimización de los mismos para poder obtener una interfaz adaptable a sujetos reales y no objetos de experimento.

Abstract

Brain-computer interfaces (BCIs) suggest a breakthrough for medicine. Among its multiple uses, there is non-muscular communication for several disabled people. One of these realizations is the P300 matrix speller described by Farwell and Donchin in 1998, using the P300 ERP, studying electroencephalograms, both online and in recordings.

In this project, we intend to study different algorithms and methods for the improvement of the prediction of characters thought by a subject, so that the optimization of them is investigated in order to obtain an interface to introduce it in real subjects and not just experimental objects.

Índice

Agradecimientos	8
Resumen	10
Abstract	11
Índice	12
Índice de Tablas	14
Índice de Figuras	15
1 Introducción	19
2 El cerebro y el EEG	22
2.1 <i>Fisiología cerebral</i>	22
2.1.1 División del cerebro	22
2.1.2 Funcionamiento eléctrico	24
2.2 <i>El Electroencefalograma (EEG)</i>	26
2.2.1 Funcionamiento	26
2.2.2 Características principales	28
2.2.3 Sistema de sensores	30
3 ERPs	35
3.1 <i>Características</i>	35
3.2 <i>Referencia de potencial</i>	36
3.3 <i>Detección</i>	37
3.4 <i>Onda P300 o P3</i>	38
3.4.1 Paradigma oddball	40
3.5 <i>BCI</i>	41
3.5.1 Donchin Speller	42
4 Experimento	45
4.1 <i>Base de datos</i>	45
4.1.1 Ficheros de datos Matlab	47
4.2 <i>Librería wrym</i>	49
4.2.1 Módulos	49
4.2.2 Contenido	50
4.3 <i>Fichero de ejemplo</i>	51

4.3.1	Procesado de los datos	52
4.3.2	Dibujo de los datos	53
4.4	<i>Algoritmo LDA</i>	53
4.4.1	Terminología	53
4.4.2	Algoritmo y probabilidades	54
4.5	<i>Presentación de resultados</i>	55
5	Conclusiones	62
5.1	<i>Predicción de caracteres</i>	62
5.2	<i>Detección de ondas en diferentes canales</i>	63
5.3	<i>Gráficas cerebrales</i>	64
5.4	<i>Generalidades</i>	64
6	Líneas futuras	66
6.1	<i>Investigación actual</i>	66
6.1.1	Speller mediante un ECoG	66
6.1.2	Nuevos conceptos del Speller	67
6.1.3	Estudio de diferentes algoritmos de diferenciación	68
6.2	<i>Líneas futuras en base a las conclusiones obtenidas</i>	69
7	Bibliografía	71
	Anexo a: códigos utilizados	77

ÍNDICE DE TABLAS

Tabla 1. Tipos de encefalografía	25
Tabla 2. Bandas frecuenciales en el EEG	27
Tabla 3. Correspondencias de electrodos en el EEG con los lóbulos	29
Tabla 4. Referencias de posición en el EEG	30
Tabla 5. Tipos de BCI	39
Tabla 6. Variables de la Base de Datos	45
Tabla 7. Comparación de resultados correctos y predichos, sujeto A	54
Tabla 8. Comparación de resultados correctos y predichos, sujeto B	54

ÍNDICE DE FIGURAS

Figura 1. Divisiones principales del cerebro	20
Figura 2. Lóbulos de la corteza cerebral	21
Figura 3. Partes de una neurona	22
Figura 4. Transmisión de señales entre neuronas	23
Figura 5. Suma de potenciales (PIPS y PEPS) en una neurona	24
Figura 6. ECoG	25
Figura 7. E-EEG	26
Figura 8. Ejemplo de ritmos del EEG	28
Figura 9. Casco de malla para encefalografía	29
Figura 10. Posicionamiento 10/20 de un sistema de 64 canales	30
Figura 11. Distintos casos de ERPs	34
Figura 12. Context Updating Theory	37
Figura 13. SOA	38
Figura 14. Donchin Speller original	40
Figura 15. Mejora del paradigma Donchin. (a) Primer nivel. (b) Segundo nivel	41
Figura 16. Diagrama de tiempo de intensificaciones del Donchin Speller utilizado en el experimento tratado	44
Figura 17. Codificación del Speller	46
Figura 18. Contenido librería wyrm	48
Figura 19. Contenido directorio “wyrn”	48
Figura 20. Contenido directorio “examples”	48
Figura 21. IDE Spyder	49
Figura 22. Algoritmo de generación LDA	53

Figura 23. Predicción de las letras y comparación con la secuencia correcta	53
Figura 24. Presentación de la predicción de la onda P300 en los canales FCz, Cz y Oz para el caso de aparición de target y no target, cada uno con su correspondiente leyenda	55
Figura 25. Distribución en el EEG de la amplitud de generación de la onda P300, siguiendo una evolución temporal del estímulo desde 0 ms a 700 ms aproximadamente	57
Figura 26. Representación de r^2 según canales y épocas	58
Figura 27. Mejora del Donchin Speller tradicional a un sistema audio-visual	65
Figura 28. Nuevas matrices propuestas para nuevos estudios del Donchin Speller	66

1 INTRODUCCIÓN

El cerebro es el órgano más complejo del ser humano, formando parte fundamental del Sistema Nervioso Central (SNC). Éste se encarga de todo lo que implica el funcionamiento del cuerpo y la mente, con lo que podríamos definirlo como el controlador y regulador principal de nuestro organismo. Es esencial para cualquier función vital (la respiración, el sueño...), así como otras funciones superiores (la memoria, las emociones, el razonamiento, etc).

Toda esta actividad se controla mediante las células cerebrales o neuronas que se encargan de recibir, procesar y transmitir la información utilizando impulsos electro-químicos mediante conexiones de multitud de neuronas. Estos impulsos, o actividad eléctrica, no son más que los restos del intenso flujo de la actividad cerebral, es decir, de toda la información que nuestro cuerpo procesa, tanto fisiológica como sensorial (estímulos, órdenes...).

Debido a que la naturaleza eléctrica de todo lo que ocurre en nuestro cuerpo radica en la carga y descarga de las células cerebrales (Chornet y Oteo 2002) se plantea la posibilidad de su detección mediante electrodos y se ejecuta en lo que se conoce como el electroencefalograma (EEG), actualmente una prueba no invasiva que se realiza sobre el cráneo.

A partir de este momento, se comienzan a estudiar las ondas que se registran en el EEG. Esto marca un antes y un después en la neurología y en la psiquiatría (Saceda Corralo 2018), ya que se empiezan a distinguir los patrones de algunas de las señales, como por ejemplo las generadas por los estados de vigilia/sueño, estado de coma, etc.

Como consecuencia se deducen dos aspectos, por un lado que se puede diferenciar un EEG con un funcionamiento normal del SNC de uno patológico (para lo que es comúnmente usado) y por otro que un estímulo hace que las neuronas emitan corrientes eléctricas de manera sincronizada para procesarlo. A pesar de ser fácilmente confundidos con el resto de ondas del EEG, se demuestra que es posible detectarlos. De esta manera definimos los potenciales relacionados con eventos o ERPs (Event Related Potentials).

Estos se nombran según el tiempo que tardan en aparecer tras el estímulo y de si el potencial que aparece es positivo o negativo frente a un sistema de referencia o “potencial cero”, definido previamente. Cada una de estas señales se relacionan con un tipo de excitación determinada, por lo que se comienzan a utilizar estos ERPs para diversas aplicaciones.

En este trabajo nos centraremos especialmente en uno de estos potenciales evocados, la onda P300, que definiremos con más extensión más adelante. Su aparición surge cuando se introducen estímulos inesperados para el sujeto que son relevantes para la tarea que debe realizar (Núñez-Peña, Corral y Escerra 2004), pudiendo provocarlo mediante el llamado “paradigma oddball”.

Todo esto pone de manifiesto la infinidad de posibilidades que nos ofrece la identificación de ondas en el EEG.

De esta manera, gracias a los grandes avances, se plantea el uso de la tecnología y la computación para el procesamiento de estos datos, dando lugar a una interfaz cerebro-computadora (BCI) de forma que tendremos la posibilidad de comunicarnos con un ordenador mediante las señales cerebrales y convertirlas en comandos para los mismos (Vaid, Singh y Kaur 2015). La consecuencia de ser capaces de “leer” lo que quiere decir nuestro cerebro es tener la posibilidad de ayudar a los pacientes que no pueden comunicarse o que no son capaces de interactuar por sí mismos debido a su enfermedad, mediante una técnica no invasiva como es el EEG (Machado et al. 2010).

En el proceso de la aplicación de este concepto a la clínica, se estudia la posibilidad de aplicarlo para ayudar a las personas que no tienen capacidad comunicativa a hacerlo mediante una pantalla de ordenador conteniendo todas las letras y otros símbolos (Farwell y Donchin 1988). Llamaremos a esta tecnología el “Donchin Speller”, objeto de investigación durante años y en la actualidad.

Así pues, nos lleva al experimento que trataremos en el trabajo, cuyo objetivo es el estudio de una BCI (interfaz cerebro-computadora) que procesa electroencefalogramas con potenciales evocados provocados mediante el paradigma anteriormente nombrado.

Analizaremos el funcionamiento de una de las librerías escritas en lenguaje Python utilizando un ejemplo de su uso, describiendo el tratamiento que reciben los datos, la precisión que se consigue en experimento, los algoritmos utilizados y otros aspectos sobre el análisis de esos mismos resultados.

2 EL CEREBRO Y EL EEG

El Sistema Nervioso Central (SNC) se compone de todo el entramado de tejidos que captura y procesa estímulos para una interacción eficaz con el medio. En él, el cerebro es la parte más importante, ya que en él se procesa todo lo que ocurre en nuestro organismo, siendo el encargado de las funciones vitales, motoras, el tratamiento de la información captada desde los sentidos, las emociones, el comportamiento, entre otras muchas.

Lo que permite principalmente esta función son las neuronas, que son las células especializadas, interconectadas y transmitiéndose toda la información y procesándola entre ellas. Éstas se comunican de manera electroquímica, es decir, mediante impulsos nerviosos llamados “potenciales de acción”, haciendo este intercambio exitoso mediante la sinapsis, donde las neuronas se ponen en contacto mediante descargas e impulsos eléctricos.

2.1 Fisiología cerebral

2.1.1 División del cerebro

El encéfalo se divide en tres partes principales: prosencéfalo, mesencéfalo y el romboencéfalo, subdividiéndose las mismas en otras muchas partes (Bailey 2018a).

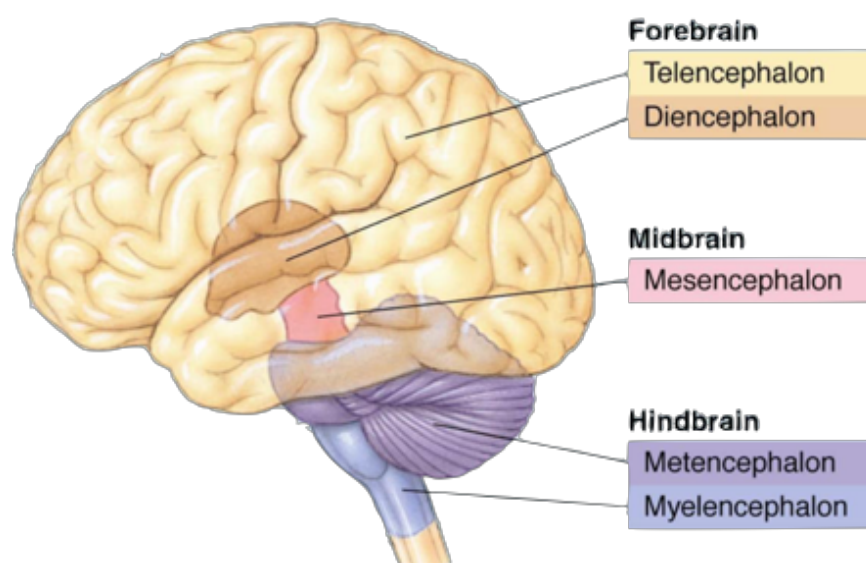


Figura 1. Divisiones principales del cerebro

Todo el cerebro es partícipe del procesamiento de la información, así como lo es el resto del Sistema Nervioso Central, por lo que nos centraremos en posicionar los elementos que recogen la actividad del mismo.

Primeramente, el cerebro se divide en el hemisferio derecho e izquierdo. Como vemos en la Figura 1, el prosencéfalo (forebrain), se divide a su vez en el telencéfalo y el diencefalo. Nos centramos en el telencéfalo, que ocupa la mayor parte de la superficie del encéfalo en el que se encuentra la corteza cerebral, que es la que se subdivide en los cuatro lóbulos cerebrales, cada uno con su correspondiente función, dispuestos como en la Figura 2.

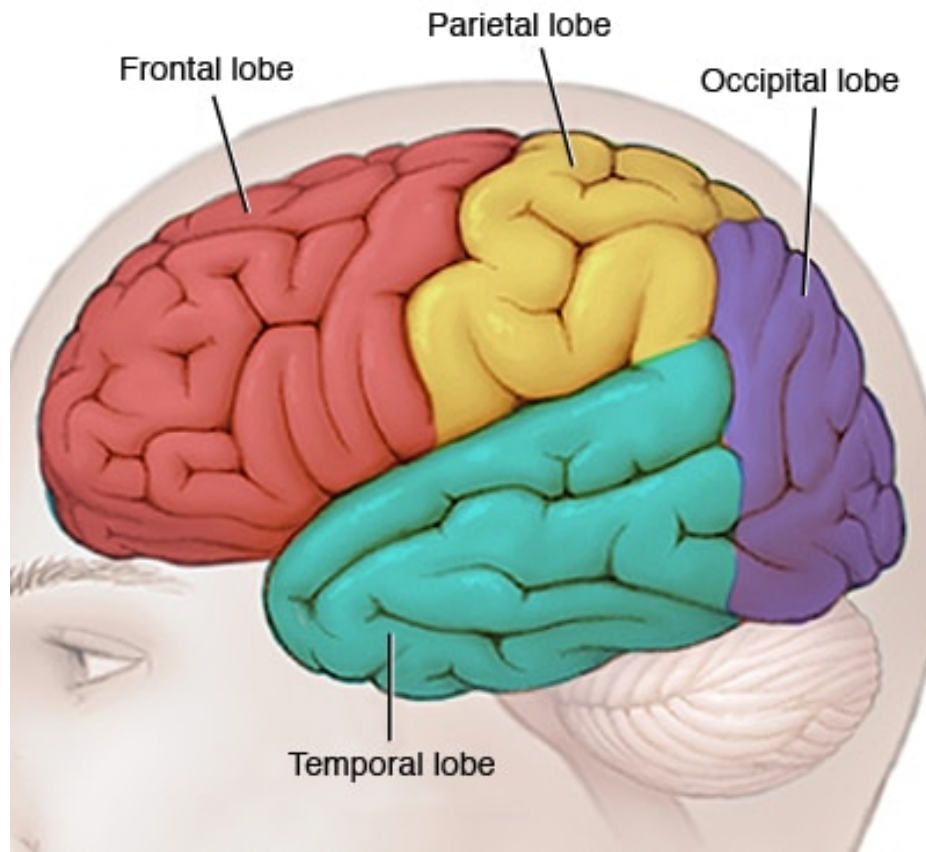


Figura 2. Lóbulos de la corteza cerebral

Las funciones de estos cuatro lóbulos cerebrales son:

- Lóbulos frontales: aquí se encuentra el área motora. Son fundamentales en los movimientos musculares voluntarios, la memoria, el pensamiento, las decisiones y la planificación. El frontal derecho se ocupa de la parte izquierda del cuerpo y el frontal izquierdo de la derecha (Bailey 2018b). Es el que más diferencia al ser humano del resto de animales.
- Lóbulos parietales: reciben y procesan la información sensorial (Bailey 2018d).
- Lóbulos occipitales: en ellos se encuentra la corteza visual primaria, que interpreta lo que recibe la retina. Estos lóbulos lo reciben, y el resto lo procesa (Bailey 2018c)
- Lóbulos temporales: se encarga de organizar la entrada sensorial, la percepción auditiva, el lenguaje, etc. También se encarga de formar y procesar mucha parte de nuestros recuerdos. Es importante destacar que en ellos se encuentra el área de Wernicke, que se encarga de la comprensión del lenguaje, así como del recuerdo de sus estructuras sintácticas, entre otras (Bailey 2017).

Todos los lóbulos cerebrales ayudan al procesamiento y tratamiento de las señales de entradas sensoriales, así como de las salidas, fundamental para el desarrollo de nuestro estudio.

Cabe decir que además de la clasificación brevemente explicada, todo ello está compuesto por las neuronas, que son las encargadas de procesar y transmitir la información inter e intracelularmente. Esto quiere decir que gracias a estas células cerebrales interconectadas, nuestro cerebro funciona como lo hace.

2.1.2 Funcionamiento eléctrico

Como hemos descrito, toda la actividad cerebral se hace mediante la interconexión de las neuronas. Estas células tienen muchas formas y tamaños, ya que como hemos visto en el apartado anterior, tienen diferentes funciones. Es decir, se especializan en almacenar la memoria o controlar los músculos, entre muchas otras tareas.

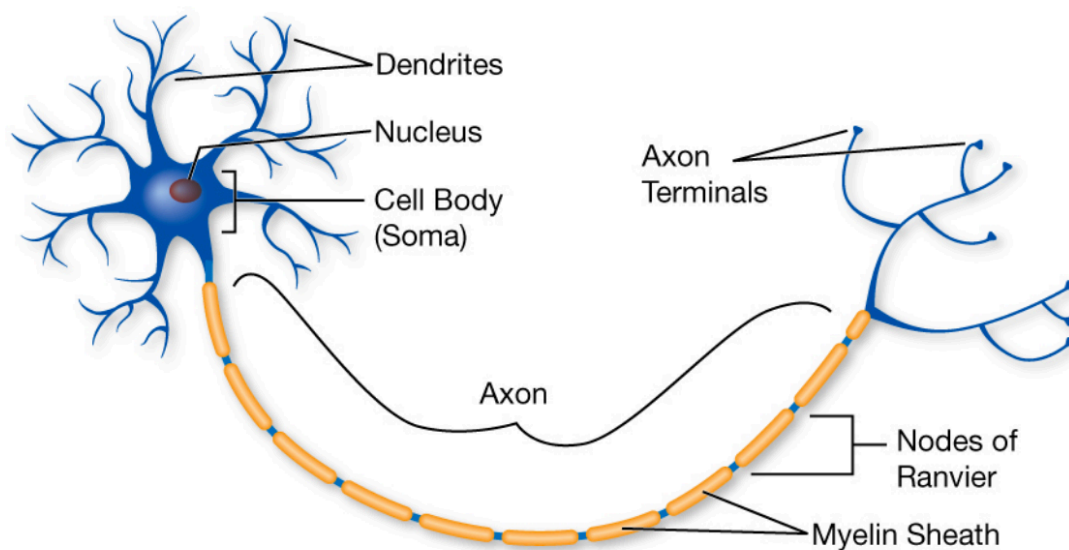


Figura 3. Partes de una neurona

Podemos ver en la Figura 3 todas las partes de una neurona. Entre ellas se encuentran (Sanchez Seisdedos 2017):

- Soma: donde reside el núcleo.
- Axones: la fibra nerviosa que permite llevar las señales eléctricas, en cuyo extremo más lejano al soma tiene terminaciones nerviosas que se conectan con las neuronas.
- Dendritas: las prolongaciones de las neuronas a través de las cuales se recibe información de otras neuronas.

Una sola neurona distingue sus propios potenciales. Se mantiene en un potencial de reposo (o voltaje a través de la membrana) o se ejecutan los potenciales de acción, que es un corto pulso eléctrico que se propaga a través de la sinapsis cuando la neurona necesita transmitir información a otra. Se inicia en el soma y se propaga por el axón de la neurona emisora o presináptica hasta llegar a la neurona receptora o postsináptica.

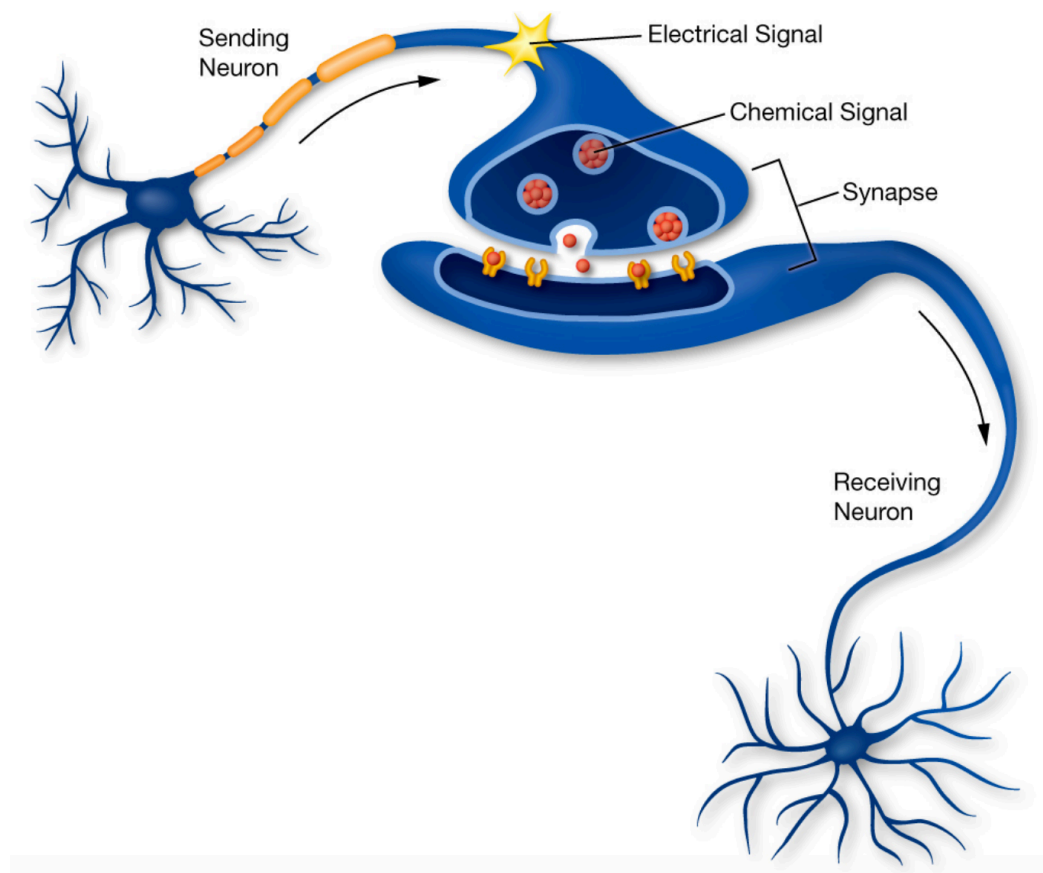


Figura 4. Transmisión de señales entre neuronas

Se transmite de manera que se inicia como una señal eléctrica, llega a la neurona postsináptica convirtiéndose en una señal química y se difunde por el hueco sináptico hasta llegar a otra neurona, en la que vuelve a convertirse en una señal eléctrica (Sörmö y Laguna 2005).

Al unirse un neurotransmisor a su receptor, se abren o cierran canales iónicos, produciendo un cambio en el potencial de membrana de la célula receptora. Hay dos opciones, que este cambio provoque un aumento o una disminución de la probabilidad de que la célula receptora dispare su propio potencial de acción. Los llamamos “potencial excitatorio postsináptico” (PEPS) o “potencial inhibitorio postsináptico” (PIPS), respectivamente.

Para que se produzca un potencial de acción en cualquier neurona cabe decir que el cambio potencial producido debe de superar un umbral, situándose este aproximadamente en -55 mV en la membrana. El potencial de reposo de una neurona aproximadamente es -70 mV , por lo que el cambio aproximado debe de ser unos $+15\text{ mV}$.

Esto puede ocurrir de dos formas: los potenciales recibidos son per se mayores que el umbral o se hace una suma, es decir, una misma neurona postsináptica puede recibir varios PEPS o PIPS de diferentes neuronas excitatorias. Esta suma puede ser espacial (potenciales postsinápticos ocurriendo en diferentes lugares pero en el mismo momento) o temporal (potenciales postsinápticos ocurriendo en el mismo lugar pero en momentos diferentes), o ambas a la vez.

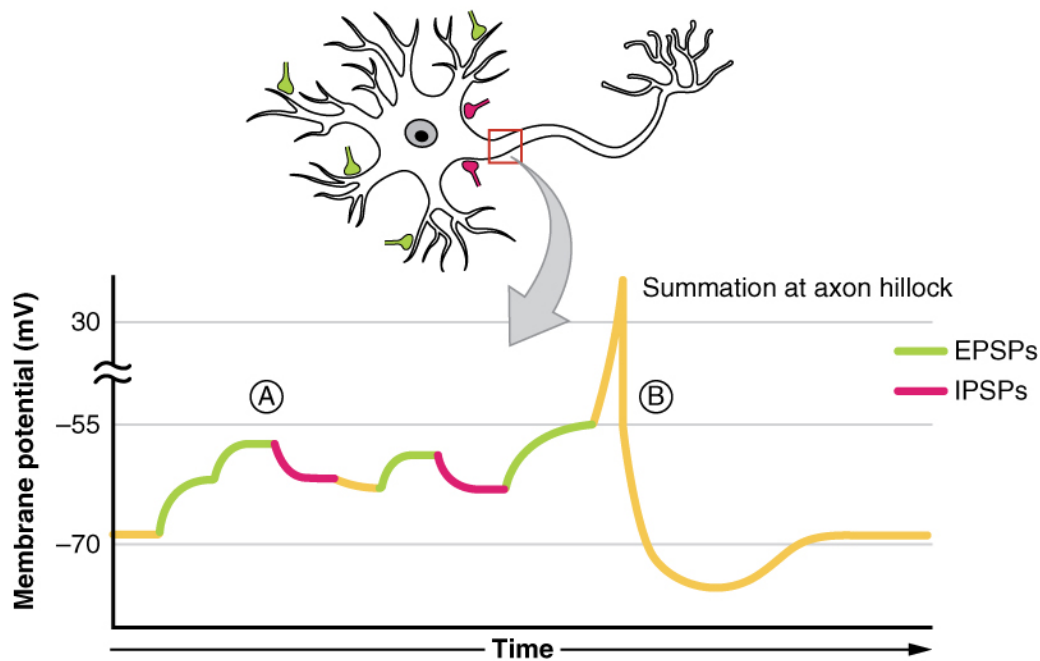


Figura 5. Suma de potenciales (PIPS y PEPS) en una neurona

Mostramos en la Figura 5 una representación de lo que puede ser la suma de diferentes potenciales, inhibidores y excitadores (OpenStax 2013).

Es de gran interés para los neurocientíficos poder hacer un registro de este flujo eléctrico, ya que se descubre que cada actividad puede dar lugar a unos tipos concretos de ondas eléctricas. A la representación de las mismas la llamamos electroencefalograma o EEG.

2.2 El Electroencefalograma (EEG)

Tal y como hemos descrito antes, el electroencefalograma (a partir de ahora EEG) es el registro de la actividad neuronal sobre la corteza cerebral, realizando un seguimiento de las ondas cerebrales producidas, por lo que con esta prueba podremos ver cómo funciona el Sistema Nervioso Central.

Se recoge mediante unos pequeños electrodos que captan esta actividad eléctrica, encargándose el aparato de amplificar esas señales, ya que son muy pequeñas, y trasladarlas a las correspondientes gráficas.

Hay diferentes electroencefalógrafos y posicionamientos de los sensores, pudiendo ser una prueba barata y no invasiva.

2.2.1 Funcionamiento

La actividad eléctrica del cerebro puede captarse de diferentes maneras, de menos a más invasivas, utilizando diferentes tipos de electrodos para cada uno de ellos y nombrándolos de manera diferente.

Tabla 1. Tipos de encefalografía

LOCALIZACION	ELECTRODOS	PRUEBA
Sobre el cuero cabelludo	SUPERFICIALES	EEG
En la base del cráneo	BASALES	
En cerebro expuesto	QUIRÚRGICOS	ECoG
En localizaciones profundas		E-EEG

En la Tabla 1, nombramos aquellos electrodos “superficiales” a aquellos que se aplican sobre el cuero cabelludo y “basales” a los que se aplican en la base del cráneo, siendo la prueba correspondiente el EEG. Cuando los electrodos son “quirúrgicos”, llamamos electrocorticograma (ECoG) si es en la superficie cerebral o estéreo electroencefalograma (E-EEG) si es dentro del mismo.

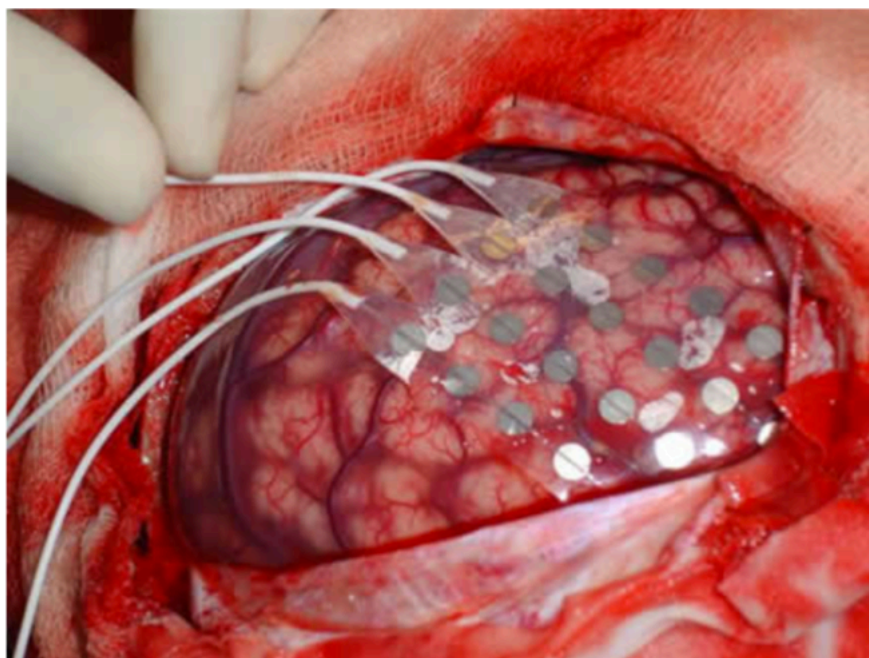


Figura 6. ECoG

Como es evidente, cada uno de estos tipos de electroencefalografía tiene su razón de ser, así como su riesgo con respecto a lo que se trate de saber. Los EEG que necesitan intervención quirúrgicas son más peligrosos, así como caros y una prueba invasiva, contraponiéndose a sus ventajas. Con el ECoG se puede hacer una estimación mucho mejor de la actividad cerebral, que el cuero cabelludo nos impide registrar con los sensores superficiales, y con el E-EEG podremos ver la actividad en puntos donde no se llega en el EEG, como el hipocampo o la amígdala (Arboleda Gómez 2011).

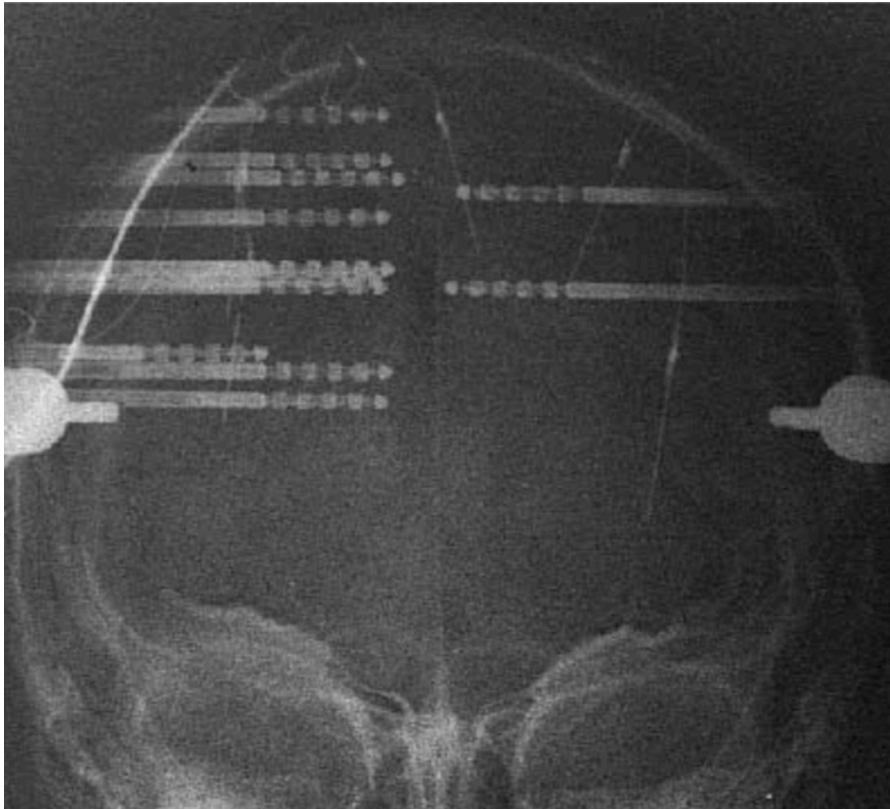


Figura 7. E-EEG

2.2.2 Características principales

Al registrar actividad eléctrica, lo que se obtienen son los potenciales lentos que se producen al sumar los PEPS y PIPS. La actividad eléctrica espontánea que se asienta en un fragmento cortical aislado se caracteriza por épocas de ondas lentas sobre las que se superponen ritmos más rápidos (Barea Navarro). Las porciones de tejido capaces de producir actividad eléctrica son los generadores:

- Generador A: A 500 μm de la superficie cortical. Produce ondas negativas.
- Generador B: A 900 μm de la superficie cortical. Produce ondas positivas. Su aparición coincide con la aparición de potenciales de acción.
- Generador C: A 900 μm de la superficie cortical. Produce ondas negativas. Su aparición coincide con la interrupción de la descarga de potenciales de acción.

Generalizando, una tensión positiva en la superficie implica una despolarización en las capas más profundas de la corteza y una negativa puede ser una hiperpolarización profunda o una despolarización superficial.

Sin embargo, la actividad neuronal no se produce en una única célula, si no que se crea una red de conexiones neuronales para transmitir y procesar la información, todas ellas participando eléctricamente. Por tanto, para que se consiga una actividad global, será necesario que las neuronas se encuentren sincronizadas. Evidentemente, cuanta más sincronización haya, más grandes y lentas son las ondas que se registran en el EEG.

Se describen diferentes tipos de ondas básicas en un EEG. Se clasifican de acuerdo a su frecuencia, su amplitud y su morfología (Trout y Martínez 2006). Para una buena interpretación, describimos las bandas frecuenciales más observables.

Tabla 2. Bandas frecuenciales en el EEG

BANDA	FRECUENCIA	AMPLITUD	DISTRIBUCIÓN	APARICIÓN
Delta	0.1 – 4 Hz	> 50 μ V		Infancia Menores de 3 meses Fase III del sueño
Theta	4 – 8 Hz	> 40 μ V	Fronto-central	Niños entre 3 meses y 5 años Fase I y del sueño Hiperventilación Fatiga
Alfa	8 – 13 Hz	15 μ V	Occipital	Consciencia
Beta	13 – 30 Hz	5 – 30 μ V		Estado de alta concentración
Gamma	25 – 100 Hz	< 5 μ V		Mecanismo de consciencia

Hacemos unas pequeñas aclaraciones respecto a la Tabla 2:

- En adultos, es anormal que aparezca la banda delta.
- En la banda Theta, una amplitud < 15 μ V es normal si se encuentra acompañada de un ritmo Alfa establecido.
- En la banda Alfa los 15 μ V son orientativos, ya que puede variar según la edad, la distancia inter electrodos, cambios en la densidad ósea...

Mostramos en la Figura 8 lo anteriormente expuesto (Ortega 2014).

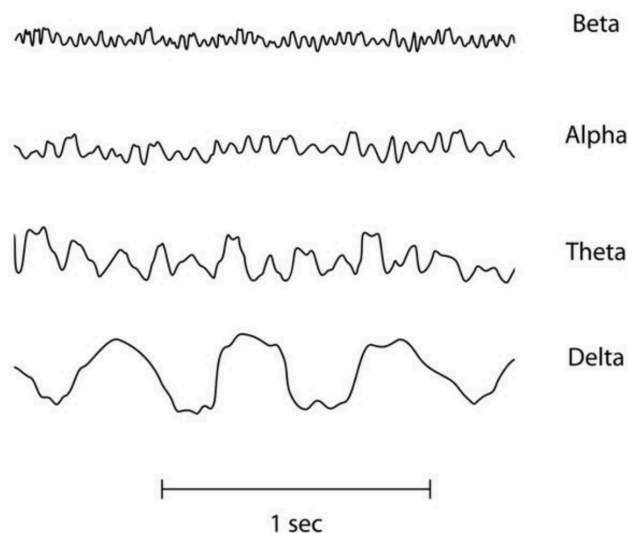


Figura 8. Ejemplo de ritmos del EEG

2.2.2.1 Artefactos

El EEG registra todas las señales eléctricas producidas por nuestro cerebro. De esta manera, tendremos señales que no se corresponderán con las anteriormente nombradas y que nos molestarán a la hora de interpretar las señales bioeléctricas.

Se definen los artefactos externos como los que no tienen que ver con el paciente (mala colocación de los electrodos, la corriente alterna, parámetros de calibración incorrectos, electricidad estática, elementos que interfieran con las señales de manera externa, que los propios electrodos estén en mal estado, etc) y los artefactos internos como los que son derivados del paciente (el electrocardiograma, un marcapasos, el pulso, la respiración, el electromiograma/artefacto muscular, movimientos oculares, el sudor, cambio en la resistencia de la piel, los movimientos del paciente, etc).

A la hora de hacer un análisis exhaustivo del EEG del paciente se hace muy necesaria la eliminación de estos artefactos.

2.2.3 Sistema de sensores

Nos centramos en los EEG con electrodos superficiales sobre el cuero cabelludo. Éstos pueden ser de diferentes maneras (Sanmillán Vidoy 2017):

- Adheridos: Discos metálicos de $\varnothing = 5$ mm. Se adhieren con pasta conductora y se fijan con un aislante (colodión). Dan resistencias de contacto bajas (1-2 K Ω).
- De contacto: Pequeños tubos de plata clorurada roscados a soportes de plástico. En el extremo de contacto se coloca una almohadilla humedecida con una solución conductora. Se sujetan al cráneo con bandas elásticas y se conectan mediante pinzas de cocodrilo.

- Casco de malla: Los electrodos se colocan en un casco elástico, sujetado al paciente con cintas a una banda torácica.



Figura 9. Casco de malla para encefalografía

- De aguja: Uso muy limitado para recién nacidos y pacientes en la UCI.

2.2.3.1 Sistema 10/20

El Sistema 10/20 es el utilizado internacionalmente para el posicionamiento de los electrodos del EEG. Hace una correspondencia ante las localizaciones de los electrodos y el área del cuero cabelludo donde estén. Las distancias entre electrodos son el 10 % o el 20 % de la distancia total de las partes frontal-trasera o derecha-izquierda, de ahí su nombre (Ldt 2012).

Se identificarán los sensores según diferentes criterios, para así aunque se le añadan canales sean fácilmente identificables posteriormente. La ordenación será:

- Localización en el lóbulo:

Tabla 3. Correspondencias de electrodos en el EEG con los lóbulos

ELECTRODO	LÓBULO
F	Frontal
T	Temporal
C	Central
P	Parietal
O	Occipital

Hemos sombreado el lóbulo “central” porque realmente no existe y se utiliza como herramienta de identificación.

- Hemisferios:
 - Derecho: números pares
 - Izquierdo: números impares.
- Para identificar la línea media, se le añade una “Z” (zero).

Por último, necesitamos añadir los puntos de referencia (Talamillo García 2011).

Tabla 4. Referencias de posición en el EEG

REFERENCIA	POSICIÓN
Nasión	Punto frente-nariz
Inión	Protuberancia occipital
Punto preauricular	Delante del trago ¹ de cada oreja

Debemos añadir un par de aclaraciones. Por una parte, muchas veces es necesario añadir letras acompañantes del sistema anteriormente descrito, ya que para los EEG con más sensores es necesario describir más “lóbulos” o posiciones, como el frontopolar (Fp). Por otra parte, los electrodos “A” son los de las orejas.

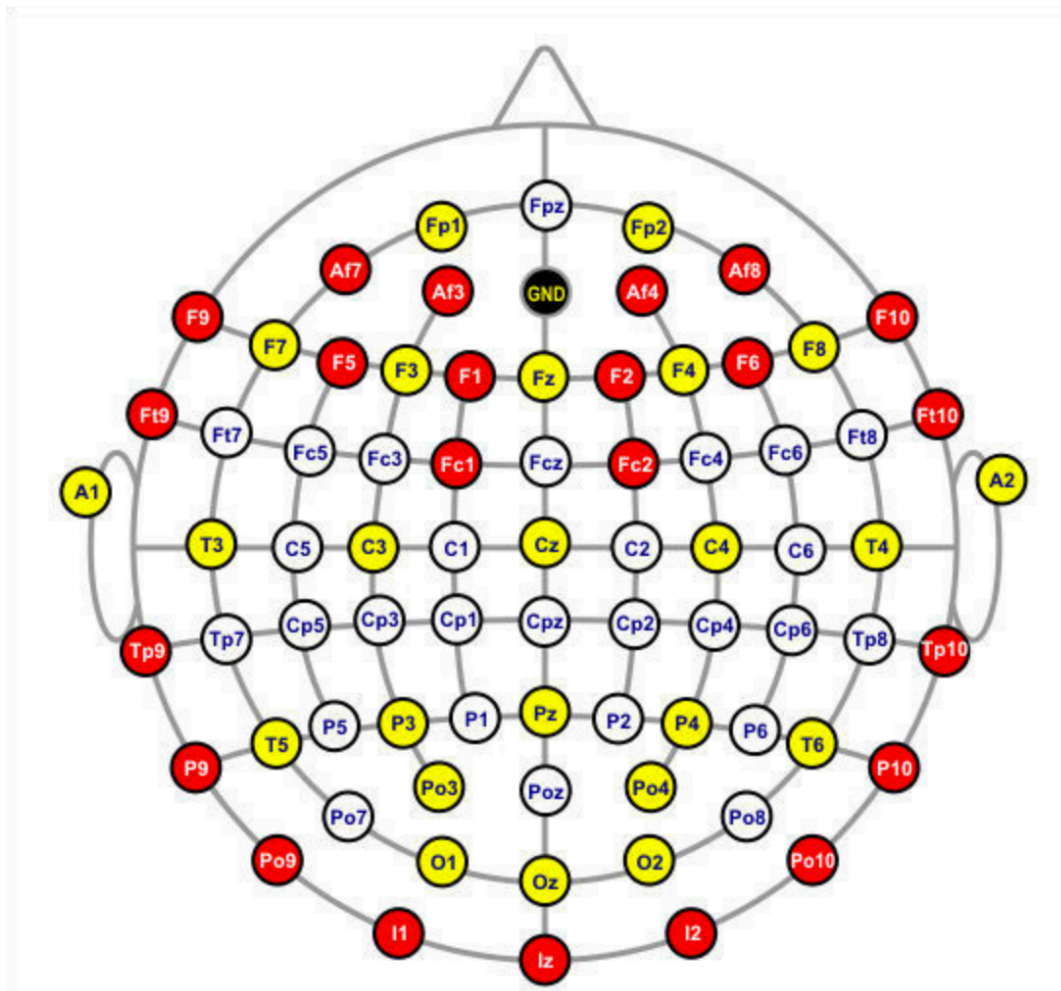


Figura 10. Posicionamiento 10/20 de un sistema de 64 canales

¹ Zona de tejido cartilaginoso delante de la parte auricular de la oreja.

2.2.3.2 Montajes

Además de todo lo anteriormente descrito, también tenemos que hacer una mención al montaje utilizado, ya que cada uno de los electrodos corresponde a un canal o punto de medida del EEG, y por lo tanto hacer una buena interpretación de los voltajes del mismo.

Para montarlo, los pares de electrodos se combinan de diferentes maneras. Los tipos básicos de montajes son el bipolar (transversal y longitudinal) y monopolar o referencial (Ramos-Argüelles et al. 2009). El bipolar mide la diferencia de potencial entre dos electrodos de las áreas de actividad cerebral y el monopolar calcula la diferencia de potencial, bien entre un electrodo ubicado en un área de actividad cerebral y un voltaje de referencia registrado por uno de los electrodos de referencia anteriormente descritos (por ejemplo, los de las orejas), bien entre un electrodo en zona activa y el promedio de alguno del resto de electrodos activos.

Esto tendrá que tenerse en cuenta a la hora de hacer el análisis de las señales electroencefalográficas, ya que tenemos que saber exactamente qué es lo que se está representando en el EEG para ser capaces de interpretarlo correspondiente.

En trazos generales, tendremos que considerar el sistema de referencia utilizado para nuestro estudio.

3 ERPs

Los ERPs o Event Related Potential son, como su propio nombre indica, potenciales relacionados con un evento. Esto quiere decir que dentro del EEG de fondo, ocurre una alteración en el mismo.

Estos representan una actividad neuronal que se propone de gran interés porque está temporalmente asociado a estímulos específicos (St. Louis et al. 2016). Se utilizan en la práctica clínica y en la investigación sobre el análisis de la actividad visual, auditiva, somatosensorial y otras funciones cognitivas de más alto nivel.

Como hemos descrito antes, son ondas que se relacionan temporalmente con eventos. Esto quiere decir que, aunque muy cercano temporalmente al estímulo, puede ser anterior o posterior al mismo, ya que nuestro cerebro en ciertas ocasiones es capaz de anticipar un evento.

En esto último se diferencia con los llamados Potenciales Evocados (EPs). Éstos están también relacionados con un evento, pero se producen de manera posterior al estímulo en cuestión. Es decir, primero se produce el evento y después la onda. Sin embargo, en la literatura general no se suele hacer distinción alguna entre ambos términos, por lo que en este estudio los utilizaremos indistintamente.

3.1 Características

Como ya hemos visto en los apartados anteriores, las corrientes eléctricas en el EEG se presentan cuando miles de neuronas actúan sincronizadamente para tratar alguna actividad o proceso, cuanto más sincronización mejor es el procesamiento. Por lo tanto, deducimos que cuando se produce un estímulo de cualquier tipo, la actividad neuronal se pone en marcha para procesar ese evento.

Sabemos que el EEG de fondo se encuentra entre 1 μV y 100 μV , bastante grande en comparación con el potencial que surge en los ERPs, que se encuentra entre 1 μV y 30 μV . Sin embargo, por una serie de procesamientos que posteriormente describiremos, podremos diferenciarlos del EEG de fondo.

Se pueden dar muchos tipos de detección y tests de reconocimiento en modalidades sensoriales diferentes, pruebas de respuesta tardía para medir la memoria de trabajo, GO/NOGO tests (pruebas experimentales donde hay dos tipos de estímulos: GO es el que le sucede una respuesta y NOGO es el que no requiere ninguna respuesta), entre otros muchos (Kropotov 2009). Estos distintos experimentos dan lugar a diferentes tiempos patrones de activación o inhibición de las neuronas, así como un área distinta del cerebro.

Con esto podemos definir diferentes tipos de ERPs, clasificándolos según el tiempo que tardan en aparecer, la polarización que tengan respecto de un sistema de referencia, etc. En principio los separamos en dos categorías:

- Exógenos, que aparecen en los primeros 100 ms tras el estímulo y se relacionan con el procesamiento sensorial básico. Se relacionan con las características del estímulo.
- Endógenos o cognitivos, que aparecen tras los primeros 100 ms tras el estímulo. Se relacionan con el procesamiento del estímulo del individuo (las expectativas, experiencia previa...).

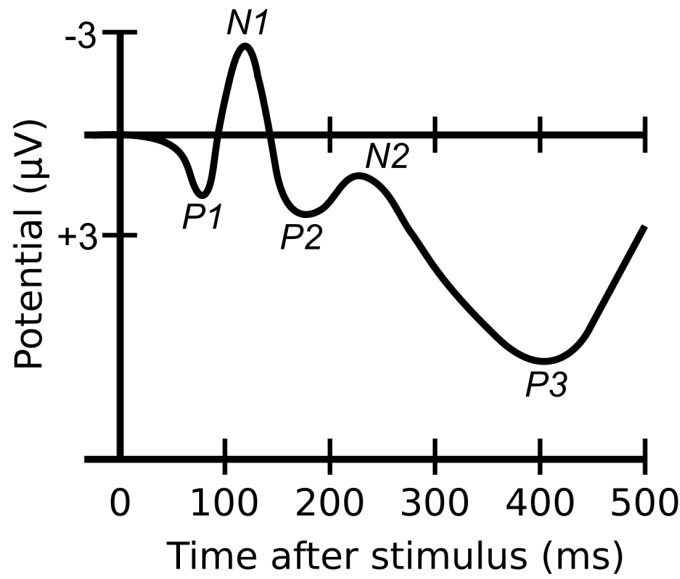


Figura 11. Distintos casos de ERPs

En la Figura 11 representamos algunos de los distintos tipos de ERPs. Vemos en la imagen que las ondas van a ser clasificadas según su tiempo de aparición, pero también por la polarización que tienen según un sistema de referencia, en este caso la tensión tomada en la apófisis mastoide (detrás de la oreja).

Nombraremos las ondas según su polaridad (P o N, positiva o negativa respectivamente) y después un número que corresponde al tiempo que tarda en aparecer, siendo este en ms (P100, N100...) o numerándolas (P1 es la primera onda que aparece positiva, N1 la primera onda que aparece negativa, etc). Cada una corresponde a un tipo de estímulo presentado diferente (por ejemplo, la N3 se relaciona con la incongruencia semántica en una frase), así como su mayor amplitud se obtiene en zonas determinadas del cerebro.

Caracterizando estos potenciales, podemos decir que se relacionan de manera que el ERP varía la amplitud con la latencia, por lo que cuánto más inesperado sea, mayor será la amplitud, así como cada área del cerebro lo procesa y el tiempo que tarda en aparecer, que se relaciona con la complejidad de proceso del estímulo.

3.2 Referencia de potencial

Es complicado elegir una referencia de potencial correcta para la óptima detección de los ERPs, debido a que en nuestro cuerpo no hay ningún punto donde no haya actividad neuronal, ya que recibe constantemente señales eléctricas que podrán ser captadas por los sensores. No existe ninguna localización donde el potencial puede ser considerado como completamente neutral (Lei y Liao 2017).

Tenemos distintas opciones que han sido estudiadas para saber cuáles son mejores o peores en la actuación. Una de las opciones mayormente aceptadas es considerar que el valor promedio de todos los canales que se registran es un valor “virtual” potencial nulo. Para que esto sea válido, es necesaria una gran densidad de sensores, así como una cobertura total de la superficie de la cabeza.

Siendo N el número total de electrodos del EEG y $v_n(t)$ la tensión absoluta medida correspondiente al n -ésimo electrodo, definimos la tensión promedio:

$$V_{av} = \frac{1}{N} \sum_{n=1}^N v_n(t) \quad (1)$$

De esta manera, el aparato calcula una tensión normalizada en cada uno de los electrodos del EEG:

$$\bar{v}_n(t) = v_n(t) - V_{av} \quad (2)$$

Comprobamos que:

$$\sum_{n=1}^N \bar{v}_n(t) = 0 \quad (3)$$

Como vemos en la ecuación (3), al anularse la suma de las tensiones normalizadas, tiene que haber diferencia de signo entre las diferentes ondas. Con ello deducimos que el signo deja de ser algo identificativo de las ondas, así también de los ERPs, ya que en algunos sensores se encontrarán con signo positivo y en otros negativos.

3.3 Detección

Hemos explicado previamente que los ERPs están “escondidos” bajo el ruido de fondo del EEG, por lo que tenemos que buscar una manera de eliminar esas señales de fondo para detectar el potencial evocado.

En base a las características anteriormente descritas, tanto del EEG como de los propios ERPs, se deduce que hay diferentes pasos para una óptima localización del mismo.

- 1) Eliminación de los registros del EEG que estén defectuosos, ya que serán inservibles y entorpecerán la detección.
- 2) Eliminación de las componentes frecuenciales que no nos interesan. Se hará un filtrado paso de banda de las señales del EEG para quitar el ruido de fondo de alta frecuencia y nos deje unos registros más limpios. También se elimina la componente DC.
- 3) Eliminación de los artefactos que queden dentro de la banda de interés (ECG, EOG...). Se convierte en un problema de separación de fuentes para la distinción de las diferentes componentes dentro de un EEG. Uno de los algoritmos más utilizados es el ICA (Independent Component Analysis).

Una vez nos hayamos quedado con una señal más o menos limpia, procederemos a la identificación de los ERPs.

- 4) Se divide el electroencefalograma en **épocas**. Como el EEG tiene un gran ruido de fondo, se hace un experimento donde se repita varias veces el mismo estímulo, separándolos por un intervalo de tiempo determinado. Cada uno de estos intervalos son las épocas.
- 5) Se hace un promediado de las señales del EEG con todas las épocas. Clásicamente, la contribución del ruido del EEG en el promediado del ERP debería tender a cero (Yeung et al. 2004) cuantas más épocas se promedien juntas.

Si se presenta a una persona un mismo estímulo que provoque un ERP M veces, en $\{t_1, t_2, \dots, t_n, \dots, t_M\}$, definimos $v(t_n, t)$ como la tensión de un electrodo cualquiera en un tiempo alrededor de t_n .

$$v(t_n, t) = v(t_n + t), \text{ para } t \in (-\Delta_1, \Delta_2) \quad (4)$$

Definimos $s_n(t_n, t)$, siendo la señal del ERP, que también va a depender de la época, y $e_n(t_n, t)$ la correspondiente al ruido de fondo del EEG. Se puede separar la señal medida en los electrodos en las diferentes épocas como:

$$v_n(t_n, t) = s_n(t_n, t) + e_n(t_n, t) \quad (5)$$

Así, obtenemos:

$$\begin{aligned}\bar{v}(t) &= \frac{1}{M} \sum_{n=1}^M (s_n(t_n, t) + e_n(t_n, t)) \\ \bar{v}(t) &= \frac{1}{M} \sum_{n=1}^M s_n(t) = \bar{s}(t)\end{aligned}\tag{6}$$

Y, con ello, la señal del ERP promediada, $\bar{s}(t)$.

Para obtener una optimización de detección de la señal promediada del ERP, podemos tener en cuenta la sincronía que se produce a la hora de su aparición tras el estímulo, por lo que se puede mejorar al hacer un reset de la fase del EEG (Yeung et al. 2004).

3.4 Onda P300 o P3

La onda P300 ha sido y está siendo uno de los potenciales evocados más estudiados. Se trata de una componente positiva de los ERP cuyo pico aparece unos 300 ms después del estímulo, aunque puede que se alargue hasta 900 ms, con una amplitud máxima de 10 μ V que se encuentra en la parte parietocentral. Se clasifica como **endógeno**, ya que está estrechamente relacionado con el procesamiento del estímulo.

No hay causa inequívoca de esta onda, aunque hay muchas teorías con respecto al tema. Por una parte, se relaciona con que al aparecer un estímulo, el cerebro lo “guarda” primero en la inconsciencia y lo pasa posteriormente a la consciencia, siendo esta transferencia de información la P300 en sí (Picton 1992). Por otra parte, también se relaciona con los mecanismos de cambio del modelo de actuación del sujeto receptor del estímulo, adaptando la reacción del mismo a la nueva situación. Tanto la atención como la memoria se involucran en estos procesos, ya que el cerebro compara entre estímulos y diferencia el que no es igual que el anterior, cambiando así el patrón de comportamiento (Linden 2005).

A esto se le llama **Context Updating Theory**, ya que la onda P300 indexa las actividades del cerebro que subyacen en la actualización de la representación mental que se tienen por los estímulos entrantes. Se produce una entrada sensorial inicial, posteriormente se realiza una comparación que dirige la atención del nuevo estímulo con el evento anterior de la memoria de trabajo. Si no se detecta ningún cambio, se mantiene el esquema del estímulo y sólo se evocan los potenciales sensoriales. Si se detecta un nuevo estímulo, se produce una actualización de la representación y se produce la onda P3 (Somani y Shukla 2012). Lo mostramos de manera esquematizada en la Figura 12.

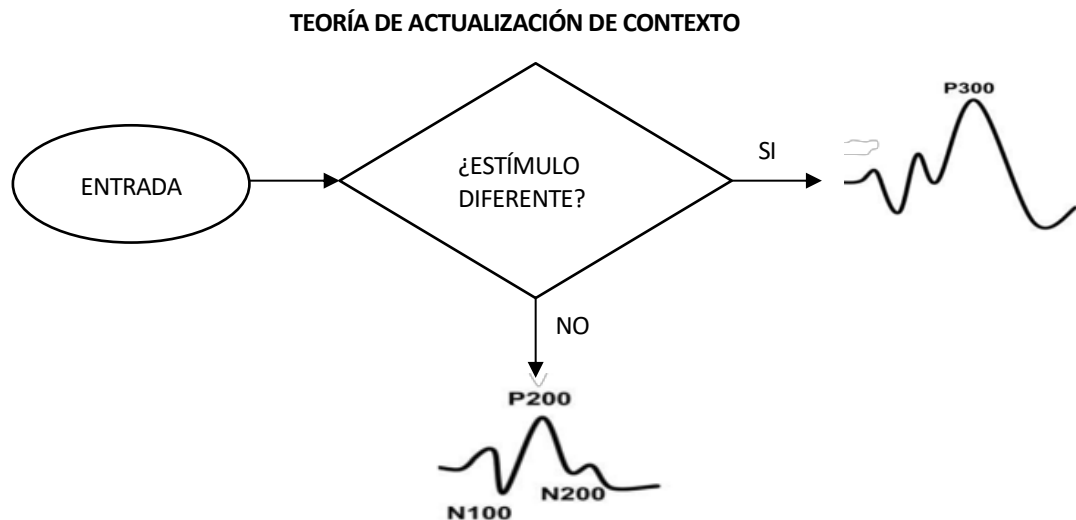


Figura 12. Context Updating Theory

Como hemos referido antes, la onda P3 se genera al distinguir un estímulo diferente dentro de un grupo de estímulos iguales. De ahí podemos derivar en dos casos (Soltani y Knight 2000):

- P3a: Se propone como la onda que aparece rápida y de manera involuntaria cuando se hace un cambio de atención a un estímulo inesperado. Tiene una distribución fronto-central, se presenta unos 60-89 ms antes y se habitúa rápidamente de las primeras 5-10 presentaciones del estímulo, por lo que acaba por desaparecer.
- P3b: Aparece cuando al sujeto se le presenta una tarea de reconocimiento de un estímulo diferente dentro de unos iguales (Polich y Kok 1995), de manera que éste se encuentra concentrado y atendiendo a que esto pueda ocurrir en cualquier momento. Nos centraremos en esta onda en particular, que no la distinguiremos a lo largo del trabajo como P3.

Todo lo que sabemos sobre esta onda nos da lugar a pensar sobre cómo podemos provocarla, así como los diferentes usos que podemos hacer de la misma.

Se proponen experimentos en los que el sujeto tenga que identificar un target que vaya apareciendo con cierta probabilidad, registrando así la onda. Estudiaremos con más detalle exactamente cómo provocarla más adelante.

Esto lleva a estudiar diversas aplicaciones. Primeramente, se descubre que hay afectaciones de la onda P3 en diferentes enfermedades, viéndose en pacientes con esquizofrenia, autismo, demencia, alcoholismo, diferentes lesiones, entre otras (Picton 1992), por lo que se puede hacer un experimento de generación de esta onda para el diagnóstico de las correspondientes afecciones. Por otra parte, se estudia cómo utilizar la forma de aparición de la onda para poder aplicarlo, como por ejemplo en el Donchin Speller (que explicaremos con más detalle más adelante), aplicaciones en personas discapacitadas para su ayuda, etc.

3.4.1 Paradigma oddball

Se define este paradigma como una de las tareas psicológicas que genera la onda P3. En este experimento se distinguen dos tipos de estímulos: *target* (diana) y *no target*. El elemento *target* es aquel que el individuo ha de reconocer y los *no target* son el resto.

Con esto, se le presenta al sujeto una serie de estímulos iguales (*no target*), entre los que se intercala un estímulo diferente (*target*) que el individuo tiene que reconocer y realizar una acción respecto al mismo, de esta manera manteniendo la atención (contando las veces que aparece el *target*, presionando un botón...), no haciendo nada en el resto de estímulos.

Se demuestra que cuanto más probabilidad de aparecer tenga el *target*, menos amplitud nos encontramos en la onda P3, hipotéticamente porque la memoria inmediata “decae” y se va restaurando por los eventos neuronales que se producen en la presentación del nuevo estímulo *target* (Polich y Margala 1997).

Sin embargo, tenemos que señalar diferentes aspectos.

En psicología experimental se define el “SOA” (Stimulus Onset Asynchrony), que denota la cantidad de tiempo desde que se presenta un estímulo S1 y el siguiente S2. Con esto se pretende comprobar el efecto de un estímulo sobre el otro. Se presentan dos opciones:

- Si el SOA es muy bajo se produce un solapamiento de ambos estímulos.
- Si el SOA es muy alto, el efecto del primer estímulo puede haber cesado y el S2 se estudia como un estímulo aislado.

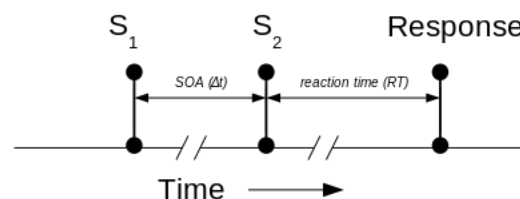


Figura 13. SOA

Podemos hacer una analogía con este experimento en el concepto Target-To-Target Interval (TTI) y cómo afecta a la presentación de la P3 (Citi, Poli y Cinel 2010). Es decir, cuánto tiempo pasa desde que aparece un *target* al siguiente. Es necesario que se deje un *Recovery Time* entre ambos, ya que si, por ejemplo, aparecen dos *target* seguidos, el cerebro no tiene recursos para hacer un ERP con una amplitud considerable, teniendo así efecto directo con la amplitud de la P3.

Así pues, se ha de tener en cuenta a la hora de realizar el experimento, ya que además se debe mantener un equilibrio entre no hacerlo demasiado ajustado ni demasiado largo, ya que la latencia aumentará y será más complicado de identificar. Además, en muchas de las aplicaciones que utilizan este tipo de potencial evocado, es necesaria cierta rapidez, por lo que es complejo un equilibrio entre mantener una probabilidad baja y el tiempo total utilizado.

También existen otros tipos de paradigmas para generar la P3, como el paradigma S1-S2, los paradigmas de detección de señal, etc, aunque son menos utilizados que el presente (Fabiani, Gratton y Karis 1987).

3.5 BCI

Las BCIs (Brain Computer Interface) son unos sistemas de comunicación, tanto hardware como software que permite al humano interactuar con su alrededor sin la necesidad de nervios ni músculos periféricos, sino únicamente usando señales de control que se generan mediante el EEG (Saddique y Siddiqui 2009).

Es un sistema de inteligencia artificial que puede reconocer una serie de patrones en las señales cerebrales, siguiendo en general diferentes etapas:

1. **Adquisición de la señal.** Captura las señales cerebrales y en muchos casos reduce el ruido y procesa los artefactos.
2. **Preprocesado o mejora de la señal.** Prepara la señal para el resto de etapas.
3. **Extracción de las características.** Identifica información que se ha recogido en el EEG.
4. **Clasificación.** Se mapea la señal en un vector que contiene características efectivas y discriminantes del resto de señales observadas. Es la etapa más complicada, ya que como sabemos las señales cerebrales están completamente mezcladas en el EEG. También el vector debe de ser de un orden bajo, ya que si no se complica la interfaz, aunque con cuidado de no perder información necesaria.
5. **Interfaz de control.** Traduce las señales clasificadas en comandos que controlarán cualquier dispositivo conectado.

La primera BCI que se sabe que fue utilizada, la utilizaron Farwell y Donchin, siendo la misma basada en la onda P3, el Donchin Speller. También se ha estudiado el control de la BCI mediante el uso de los ritmos motores-sensoriales (SMR). Otro tipo de controlador de BCI utiliza los potenciales evocados visuales en estado estacionario (SSVEPs) (McFarland y Wolpaw 2017).

Esto nos da lugar a una clasificación de las BCI que se muestran en la Tabla 5. El uso de cada una de ellas será dependiente de la aplicación utilizada.

Tabla 5. Tipos de BCI

ENFOQUE	SEÑALES CEREBRALES	VENTAJAS	DESVENTAJAS
Exógeno	SSVEP	No necesita formación	Atención permanente a los estímulos externos que puede causar cansancio
	P300	Fácil configuración Tasa de bit 60 b/min 1 canal del EEG requerido	
Endógeno	SCPs (Slow Cortical Potentials)	Independiente de estimulación Opera a voluntad	Mucho tiempo No todos los usuarios lo controlan Registro multicanal Baja tasa de bit (20-30 b/min)
	SMR	Para personas con órganos sensoriales afectados Para control de cursor	

Su uso cada vez está más extendido, ya que éstas ofrecen una nueva manera de comunicación y control sin ninguna intervención de los músculos o nervios periféricos, por lo que puede servir mucho de ayuda a las personas con discapacidad motora.

También se debe a que se han hecho numerosos estudios y avances en el campo de las BCI basadas en el EEG, que producen una buena calidad de señal, así como un método no invasivo y un equipamiento muy fácil de usar y montar.

Esto es muy poderoso, ya que poco a poco las aplicaciones basadas en las BCI representan una gran herramienta para revelar información del cerebro del sujeto que no es expresada.

3.5.1 Donchin Speller

Nos vamos a centrar en las BCI que utilizan la P300, ya que son muy ventajosas debido a que esta onda, como sabemos, se genera espontáneamente y como consecuencia de un estímulo, sin necesitar ningún entrenamiento previo por parte del sujeto.

Una de las más conocidas es el Donchin Speller, que sirve para que las personas sean capaces de deletrear palabras, deducidas mediante una BCI basada en la onda P3 que se genera en el experimento.

Uno de los más utilizados y conocidos fue propuesto por Farwell y Donchin en 1998. Tal y como fue descrito por ellos, el sistema funciona de tal manera que se muestra en una pantalla controlada por un ordenador una matriz de 6x6 que contiene todas las letras del alfabeto, así como algunos símbolos (Farwell y Donchin 1988).

Se basa en el paradigma oddball previamente explicado. Los estímulos que aparecen en el experimentos son las intensificaciones mediante un flash de una columna o fila de la matriz. Aparecen 12 eventos posibles: intensificación de 6 filas e intensificación de 6 columnas. Subdividimos éstos en dos tipos de eventos: los **no-target** y los **target**.

El sujeto que realiza el experimento piensa en uno de los caracteres de la matriz, que está en la fila i y la columna j , identificando éste como el carácter (i, j) . Identificando los eventos, el **target** sería la fila i o la columna j y el **no-target** serían todas las que no fuesen ni la fila i ni la columna j . De esta manera, cada vez que la columna o fila target se intensifique, se producirá la onda P300.

A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	1	2	3	4
5	6	7	8	9	-

Figura 14. Donchin Speller original

En el experimento dado se obtiene una tasa y precisión relativamente alta, aunque puede ser mejorada reduciendo los errores perceptivos en el paradigma Farwell-Donchin. Es decir, cuando la P3 se produce debido a flashes que no corresponden a la columna o fila target, si no a una adyacente, lo que evidentemente aumenta el error (Nicolas-Alonso y Gomez-Gil 2012).

Se propone tener regiones que se van intensificando, en lugar de filas y columnas, en una distribución de dos niveles, gráficamente en la Figura 15:

1. Primer nivel: siete grupos con siete caracteres, intensificándose de manera aleatoria. El grupo que contenga el target evocará la P3. Se ve en la Figura 15^a.
2. Segundo nivel: una vez reconocido el grupo que ha generado la onda, se centra en ese grupo, donde el procedimiento es repetido con los caracteres concretos. Figura 15b.

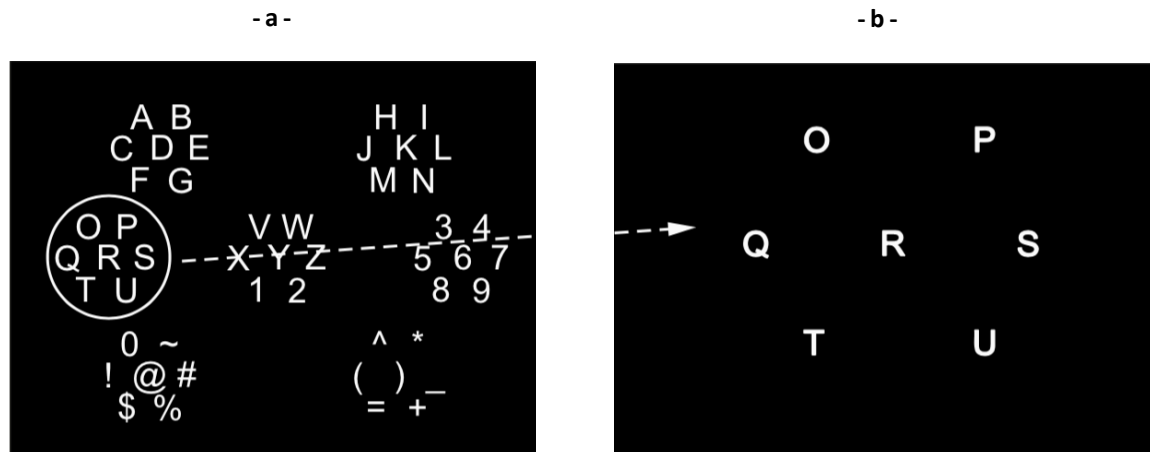


Figura 15. Mejora del paradigma Donchin. (a) Primer nivel. (b) Segundo nivel.

Hay muchos estudios y experimentos utilizando el paradigma de Donchin con el Speller, ya que es un gran método para el estudio de la onda P3, además de bastante útil. Aún sigue siendo un reto su mejora e implementación real.

En este trabajo lo utilizaremos, tratándolo experimentalmente.

4 EXPERIMENTO

En el siguiente apartado explicaremos el procedimiento seguido para la detección, caracterización, parametrización, etc, de un ERP, en concreto de la onda P3 mediante un experimento basado en el paradigma de Donchin (Blankertz y Schalk 2004).

Primeramente, hablaremos sobre la librería utilizada, describiendo los programas, el lenguaje utilizado y las funciones. Posteriormente describiremos la base de datos, así como las características de la misma, lo que contienen, cómo están grabadas las señales, cómo se han obtenido, etc. Después se detallará cómo se ha procedido a la hora de la detección de las ondas, argumentando la descripción y utilización de los códigos de ejemplo, así como su explicación.

Utilizaremos Python por diversas razones. Primeramente, a pesar de haber tenido que aprenderlo desde cero, tiene una asimilación bastante rápida ya que es muy intuitivo. También se considera el lenguaje de los investigadores por lo dicho. Por otra parte, hay mucha documentación del mismo, así como muchas librerías de gran interés para nuestro proyecto, como la que vamos a utilizar u otras como la MNE.

4.1 Base de datos²

La base de datos que vamos a mostrar y utilizar a lo largo de este trabajo fue recogida en la página web de la BCI Competition III (Schalk, Dean y Wolpaw 2004), siendo éste un concurso que pretende la validación del procesamiento de señales y clasificación de métodos para BCI, con gran relevancia para sistemas reales. Este experimento se realizó en 2004 en el Wadsworth Center (Nueva York), por el Departamento de la Salud, quienes lo organizaron.

En concreto, de todos los tipos de base de datos que tenemos disponibles, escogemos la “**Data Set II**”, correspondiente al experimento de un Donchin Speller, explicado previamente, que se clasifica dentro de los problemas multi-clase. De esta manera tendremos una grabación electroencefalográfica completa de potenciales evocados, en específico la onda P300, que han sido grabadas con la BCI2000³.

La matriz 6x6 utilizada es la misma que la mostrada en la Figura 14. Se le pidió al sujeto que se concentrase en los caracteres dados por un investigador, cada carácter en un momento diferente.

Se tratan dos sujetos en cinco sesiones diferentes. Las señales están filtradas en paso de banda con $f_{\text{inferior}} = 0.1$ Hz y $f_{\text{superior}} = 60$ Hz, posteriormente digitalizada a 240 Hz. Cada sesión consistió en un número de ejecuciones del experimento. En cada una de ellas, por cada época, sucedían diferentes pasos que explicamos de manera gráfica en la Figura 16.

² Toda la información está obtenida por el fichero de documentación de la Data Set II de la BCI Competition III (Blankertz y Schalk 2004).

³ BCI2000 es un software para las investigaciones para un sistema BCI. También puede ser usado para la adquisición de los datos, la presentación del estímulo y aplicaciones de monitorización del cerebro (NCAN)

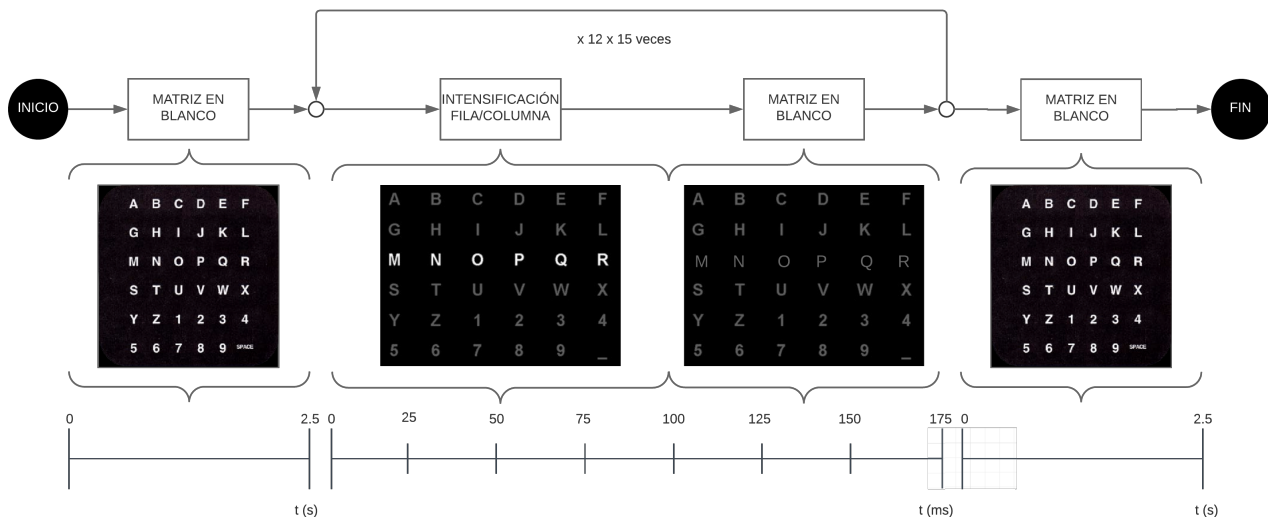


Figura 16. Diagrama de tiempo de intensificaciones del Donchin Speller utilizado en el experimento tratado.

En este caso, se dispone arriba de la matriz una sucesión de caracteres en los que el sujeto tendrá que concentrarse a lo largo del experimento. En cada época, el sujeto focalizará su atención en un carácter determinado que se indicará metiendo el carácter entre paréntesis. Por ejemplo, se dispone la serie de caracteres “ABCDE”, siendo la primera época el carácter “A”, la segunda “B”, etc, indicándole al sujeto en qué carácter debe pensar de manera que en la primera época aparecería “(A)BCDE”, en la segunda “A(B)CDE”, etc.

Como vemos, dividimos cada época en:

1. La matriz se pone en blanco durante 2.5 s antes de comenzar.
2. Se intensifica una columna o fila aleatoria durante 100 ms.
3. Le sigue la matriz en blanco durante 75 ms.
4. Se repiten los pasos 2 y 3. Como vemos indicado en la Figura 16, esto se produce 12 veces, una por cada columna (6) y fila (6). Se vuelve a repetir este bloque de repeticiones 15 veces, por lo que tendremos 12 x 15 repeticiones en total por cada carácter, es decir, 180 intensificaciones.
5. Una vez se acaben las repeticiones, se vuelve a poner la matriz en blanco durante 2.5 s.
6. Se le pide al sujeto que se concentre en el siguiente carácter, dispuesto arriba de la pantalla del speller, dejando entre paréntesis el carácter en el que correspondía fijarse.

Una vez descrito como se han obtenido los datos, indicamos específicamente lo que contiene:

- Ficheros de datos en Matlab: Subject_A_Train.mat, Subject_B_Train.mat, Subject_A_Test.mat, Subject_B_Test.mat, que ahora describiremos con detalle.
- eloc64.txt: Indicación de cómo se han colocado los 64 sensores y los nombres de los mismos.

- example.m: Fichero de ejemplo proveído.
- topoplotEEG.m: Función para dibujar un mapa topográfico de intensidad de las ondas en los sensores del EEG.

4.1.1 Ficheros de datos Matlab

Esta base de datos corresponde a la competición previamente nombrada, en la que el participante debe obtener un con la máxima precisión de predicción del carácter en el que el sujeto está pensando, así como la detección de la onda P300. Primero de todo, tenemos dos tipos de ficheros para cada sujeto:

- Los ficheros de Training: Se utilizan para entrenar al algoritmo y comprobar su buen funcionamiento, ya que contiene la sucesión de caracteres correcta.
- Los ficheros de Testing: El experimento en sí, en el que se desconoce el carácter pensado y se ha de estimar con el algoritmo de predicción.

La información se organiza en una matriz **Signal**, cuyas dimensiones son (Época x Muestras x Canales), siendo la época cada carácter de la sucesión. Por cada muestra que contiene esta matriz, tenemos unos eventos (variables) asociados que mostramos en la Tabla 6.

Tabla 6. Variables de la Base de Datos

			D1	D2
Flashing	1	Columna/Fila intensificada	Época del carácter	Muestras
	0	Columna/Fila no intensificada		
StimulusCode (*) <small>La codificación posible se muestran la Figura 17.</small>	0	Columna/Fila no intensificada		
	1,...,12	Código de intensificación de columnas/filas		
StimulusType	0	Matriz en blanco Columna/Fila intensificada no es target		
	1	Columna/Fila intensificada es target		
TargetChar	Sucesión de caracteres correctos			

Diferenciamos las variables sombreadas en gris y en azul. Las primeras las contienen todos los ficheros, pero las segundas solamente se encuentran en los ficheros de Training. Como hemos explicado previamente, estos ficheros se utilizan para la comprobación del comportamiento del algoritmo, siendo indispensables estas dos variables. Por una parte, la variable StimulusType contiene si la filas o columna que está siendo intensificada es target, por lo que se puede entrenar el algoritmo de manera que estudie cómo varía los datos de la variable Signal, formando un patrón de diferencia de intensificación target y no target. Por otra parte, tras entrenar y hacer la predicción del carácter, la variable TargetChar nos permitirá comprobar si ésta ha sido correcta.

Explicando brevemente lo contenido en la Tabla 6:

- Las variables Flashing y StimulusCode serán cero si la matriz está en blanco y serán distintas de cero cuando se produzca una intensificación. La primera nos indica simplemente si hay alguna fila o columna intensificada en ese momento con un “1” y la segunda nos indica mediante una codificación numérica que se muestra en la Figura 17 cuál es la columna o la fila concreta que está siendo intensificada en el momento.
- En la variable StimulusType se encuentra un cero si la fila o columna que se está intensificando no es target y un uno si lo es.
- En la variable TargetChar tenemos un vector con los caracteres correctos que están pensando los sujetos y que hay que predecir.

	1	2	3	4	5	6
7	A	B	C	D	E	F
8	G	H	I	J	K	L
9	M	N	O	P	Q	R
10	S	T	U	V	W	X
11	Y	Z	1	2	3	4
12	5	6	7	8	9	_

Figura 17. Codificación del Speller

Para clarificar el contenido y que quede completamente explicado, hacemos un breve resumen:

- Si Flashing = 0, StimulusCode = 0, StimulusType = 0.
- Si Flashing = 1, StimulusCode \neq 0 (dependiendo de que fila o columna se esté iluminando según la Figura 17), StimulusType = 0 ó 1 si es Target o no.

Finalizamos la explicación con unas aclaraciones, ya que es un punto clave para analizar los datos que tenemos. Por una parte, independientemente del valor de estas últimas variables, la matriz Signal tendrá sus valores concretos que corresponderán con la digitalización de todas las señales del EEG. Por otra parte, reseñar que en el Training tenemos 85 caracteres, y por tanto 85 épocas, y en el Test 100 caracteres, y por tanto 100 épocas.

4.2 Librería **wyrm**

En este trabajo utilizaremos una herramienta que nos ayudará a la detección de los ERP. Se trata de una “toolbox” basada en el lenguaje Python que ayuda a implementar una BCI, no solamente la P3 en el experimento que procederemos, sino en diversas aplicaciones como el simple análisis del EEG (Venthur et al. 2015).

Primero describiremos a grandes rasgos cómo podremos tratar los datos con la librería, ya que tenemos que saber cómo podemos sacar toda la información posible de nuestra base de datos. Para ello, se define una única estructura de datos que puede envolver todo tipo de datos durante el procesado. Se define la clase “Data”, genérica y flexible, ya que sabemos que las estructuras de datos que utilizaremos en los análisis de los EEG irán cambiando y son complejas. De esta manera, se hace una convención para la identificación.

- Datos continuos, en general los datos tal cual salen del EEG.
- Datos en épocas, que son los datos continuos separados en épocas, cada cual correspondiéndose en general a una clase. Por ello, los ejes son [clase, tiempo, canal], **clase** siempre el primero. Tiene que ser consistente con los datos continuos.
- Vectores característicos, tras el procesado de los datos. Los ejes son [clase, fv (feature vector)].

4.2.1 Módulos

Esta librería nos va a ofrecer métodos a utilizar según qué aplicación queramos darle. Hacemos una breve explicación de los módulos que tendremos, para entenderlos y utilizarlos de la manera más óptima posible, así como para saber qué posibilidades tenemos para hacer el tratamiento de los datos. Toda la documentación está obtenida de la página web de [la librería **wyrm**](#) (Venthur et al. 2015).

- **wyrm.io**: métodos para el tratamiento de entradas y salidas. Esto quiere decir que nos aporta diferentes funciones y métodos que utilizaremos para cargar o guardar datos en diferentes formatos que iremos utilizando a lo largo del procesamiento de nuestro EEG.

En este módulo encontraremos métodos para convertir de tipos de datos diferentes al tipo de datos que utiliza la librería **wyrm** (es decir, la estructura “data”), para así poder utilizar cualquier tipo de datos en nuestro programa.

En concreto, nosotros utilizaremos un método llamado `wyrm.io.load_bciomp3_ds2(filename)`, que explicaremos con más detalle más adelante.

- **wyrm.misc**: métodos misceláneos que nos ayudarán a procesar los datos.
- **wyrm.plot**: contiene métodos para diferentes tipos de dibujo de datos. Se dividen en dos tipos, los primitivos y los compuestos, ofreciendo representaciones únicas y básicas o varias primitivas y más complejas respectivamente.

Para la representación de algunos de nuestros datos, lo utilizaremos más adelante y con ello será explicado más detalladamente.

- **wyrm.processing**: métodos para el procesamiento de los datos. Este es uno de los módulos más importantes que vamos a utilizar en nuestro programa, ya que todo nuestro propósito es llegar a un resultado óptimo. Iremos explicándolo a medida que vamos utilizándolos.
- **wyrm.types**: módulo que proporciona los tipos de datos básicos utilizados en la librería, como la clase “Data”, “RingBuffer”, etc, junto con sus métodos correspondientes.

4.2.2 Contenido

Procedemos a la descripción del contenido de la librería. Contiene lo que podemos ver en la Figura 18 (Venthur et al. 2015).

doc	updated changelog	4 years ago
examples	Same experimental parameters as the online script.	4 years ago
test	updated CCA	4 years ago
wyrm	Merge branch 'master' of https://github.com/jscastanoc/wyrm into fix/...	3 years ago
.gitignore	Cleaned up .gitignore	5 years ago
.travis.yml	Removed deprecated use-mirros	4 years ago
LICENSE	Settled for MIT license:	5 years ago
Makefile	Updated target for gh_pages	4 years ago
README.md	Added Citing us block in README	3 years ago
THANKS	Added list of contributors.	4 years ago
TODO	Updated todo	5 years ago
requirements.txt	Fixed version numbers	4 years ago
setup.py	Bumped version.	4 years ago

Figura 18. Contenido librería wyrm

Los que vamos a utilizar en este proyecto se encuentran en las carpetas “wyrm” y “examples”, cuyo contenido mostramos en la Figura 19 y en la Figura 20.

__init__.py	added deprecated decorator	4 years ago
io.py	Merge branch 'master' of https://github.com/jscastanoc/wyrm into fix/...	3 years ago
misc.py	added deprecated decorator	4 years ago
plot.py	Improved plot_channels, accept epocs	4 years ago
processing.py	updated CCA	4 years ago
types.py	Improved readability	5 years ago

Figura 19. Contenido directorio “wyrm”

BCI Competition 3, Data Set 1 (mot...)	Updated mi example.	4 years ago
BCI Competition 3, Data Set 2 (P3...	Reduced filter order and updated intervals *again*	4 years ago
ERP Analysis.ipynb	Renamed filter to lfilter to avoid accidental collision with filter b...	5 years ago
ERP Classification.ipynb	Fixed typo.	5 years ago
MI Analysis.ipynb	Renamed filter to lfilter to avoid accidental collision with filter b...	5 years ago
online_experiment.py	Fixed small inconsistency	4 years ago
performance.py	Same experimental parameters as the online script.	4 years ago

Figura 20. Contenido directorio “examples”

Los módulos del directorio “wyrn” son los explicados previamente y se irán utilizando en el ejemplo que usaremos, contenido en la carpeta “examples”, concretamente en el archivo de ejemplo de la BCI Competition 3, Data Set 2 (P300 Speller).

Este es el fichero en el que nos centraremos para describir nuestro experimento y analizaremos su funcionamiento, así como los algoritmos que se usan para obtener el ERP en nuestra base de datos.

4.3 Fichero de ejemplo

Como hemos apuntado antes, utilizamos el fichero de ejemplo `speller.py` en el Anexo A para la extracción de ondas P300 de la base de datos.

Por una parte, hemos utilizado el sistema de distribución Anaconda (Anaconda Inc 2019), que nos facilitará la tarea de gestión de paquetes de Python (además de la librería `wyrn`, vamos a tener que utilizar paquetes científicos para calcular y detectar ERPs), ya que al instalarlo nos provee automáticamente de la mayoría de paquetes que se van a utilizar, así como diferentes IDE para estudiar el código. En concreto, hemos utilizado Spyder (Spyder Doc Contributors 2018) en su mayoría. Mostramos gráficamente el entorno en la Figura 21.

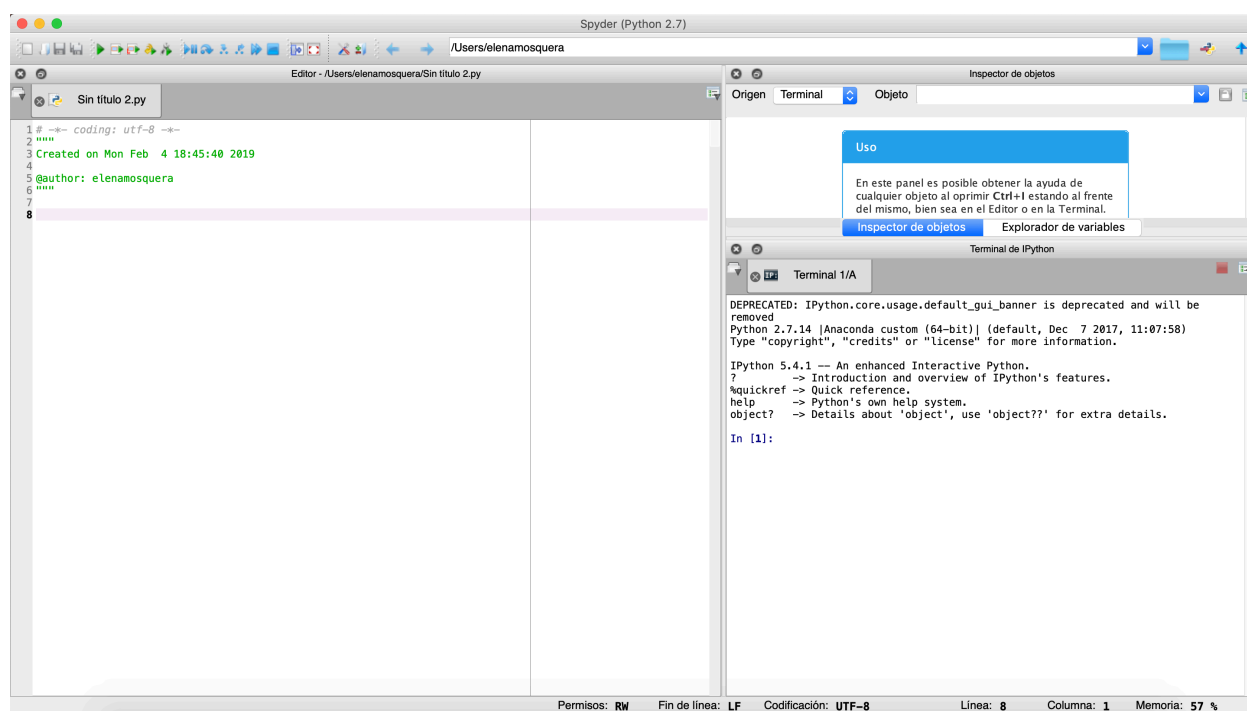


Figura 21. IDE Spyder

Como podemos observar, ofrece un editor, un analizador (inspector de objetos y variables) y una terminal de IPython. Además, como vemos en la barra superior, podemos ejecutar el código completamente, paso a paso, poner puntos de parada (es decir, debuggearlo), entre otras muchas opciones. Esto nos ha facilitado la tarea a la hora de comprender el código.

4.3.1 Procesado de los datos

Tras habernos descargado la base de datos completa de la BCI Competition III, los metemos en una carpeta a la que podamos acceder fácilmente desde la librería y cambiamos la ruta desde la que accede inicialmente el `speller.py` - ya que no es la misma que en el ejemplo.

Primeramente, se convierten los datos de tipo `.mat` al tipo de datos descrito previamente, que realmente es una convención por parte de la librería para que resulte más accesible y fácil utilizarlos en las diferentes funciones que se describen. Para ello, se utiliza el método `load_bcicom3_ds2`, perteneciente al módulo `wym.io`, que específicamente carga los datos de la Data Set II de la BCI Competition III y los convierte al formato “Continuous Data” – necesariamente la ruta donde se encuentre la BBDD debe de tener el fichero `eloc64.txt`.

Posteriormente, este ejemplo utiliza un preprocesado mediante la función `preprocessing` que logra un 91% de precisión, en los que se ordenan las señales según el fichero `eloc64.txt` según el sistema 10-20. Como sabemos, dentro de la banda del EEG que tenemos caen muchos de las señales eléctricas producidas por nuestro cuerpo, por lo que debemos de eliminar los artefactos que nos molesten para detectar el ERP.

Para ello se limpia la señal mediante dos filtros de Butterworth de orden 5, primero con un filtro Low Pass (LP) y posteriormente con un filtro High Pass (HP), con la función `scipy.signal.butter`, perteneciente al módulo `scipy`. Después se reduce la tasa de muestreo de 240 Hz a 60 Hz mediante la función `proc.subsample`, perteneciente al módulo `processing.py`, se hacen épocas de los datos y se hace el promedio de los mismos. Finalmente, se crean los vectores característicos y se da por terminada la fase de preprocesado.

Mediante los vectores característicos, se entrena el algoritmo LDA (Latent Dirichlet Allocation) – que explicaremos más detalladamente en la sección 4.4 debido a su importancia. Este entrenamiento se realiza por una función del módulo `processing.py`, `lda_train`, que nos da las probabilidades de ocurrencia, para después procesar los datos del test real y aplicarle el mismo algoritmo por la otra función `lda_apply`. Esto nos separa los estímulos según su probabilidad para saber qué tipo estamos tratando.

Tras esto, se comienza el algoritmo de predicción del estímulo en el sentido estricto.

1. Se ordenan los estímulos y los datos dados por el algoritmo LDA en matrices de 15x12, ya que como explicamos antes, el experimento hace 15 ejecuciones en bloques de 12 (6 columnas y 6 filas) que corresponden al mismo carácter.
2. Se ordenan las columnas y las filas correspondientes, de manera que la codificación de columnas y filas queden tal cual se codifica en la Figura 17.
3. Se hace la predicción del carácter según lo anterior.

Por último, se define la precisión del algoritmo mediante el feedback de los sujetos y la precisión total.

$$acc_i = \frac{N_CC_i}{l_C_i} \times 100 (\%) \quad (7)$$

$$acc_{total} = \frac{\sum_i acc_i}{2} \quad (8)$$

Siendo N_CC_i el número de caracteres iguales a la secuencia de caracteres correctos y l_C_i la longitud completa de la secuencia predicha, indicando con el subíndice “i” a qué sujeto está referida.

En la Ecuación 7 se calcula la precisión para cada sujeto y en la Ecuación 8 la total, haciendo un promediado de las precisiones individuales.

4.3.2 Dibujo de los datos

En este ejemplo también hace uso de ciertas funciones que nos mostrarán gráficamente la onda P300 que se predice, así como otros datos como un mapa topográfico de los sensores.

Para el cálculo de la onda P300 se promedian las señales en todos los aspectos. Es decir, tendremos una gran cantidad de señales que están divididas en clases (target y no target), así como 64 canales que representan el momento. De esta manera, mediante la función perteneciente al módulo `processing.py` llamada `calculate_classwise_average`, se hace el promediado. Posteriormente, sólo se representan las ondas calculadas en los canales “FcZ”, “cZ” y “oZ”, ya que son teóricamente donde mejor se ve la onda P3.

Tras esto, se define la función `plot_scalps`, que nos dibujará los sensores y en un modelado de una cabeza. De esta manera lo hará en intervalos, clasificándolos por colores según el voltaje que se registre en el sensor en cuestión. Es decir, cómo se distribuye tanto en el tiempo como en el espacio la propagación por nuestro cerebro de la onda P300 anteriormente calculada, así como lo que pasa cuando simplemente está la actividad de fondo.

Por último, también se representan en unas nuevas gráficas el valor de r^2 en una barra según las épocas, representando un índice que hace la diferencia entre los valores medios en todos los canales de los valores del EEG según si es target o no, con respecto a la longitud total de los datos. Esto nos permite crear una mayor diferencia entre máximos y mínimos y por tanto analizar mejor los datos. Se procesa mediante la función del módulo `processing.py`, `calculate_signed_r_square`, con cada época dividida.

4.4 Algoritmo LDA

Consideramos importante hacer hincapié en el algoritmo LDA, ya que es mediante el que hemos podido predecir las letras, separando los estímulos según su origen.

El método LDA trata de estimar parámetros estadísticos, surgiendo debido a la necesidad de encontrar un modelo de categorización de datos (Kolda y Bader 2009). Pretende tomar un conjunto subdivisible en tres niveles (un nivel completo que se divide en partes y a su vez éstas se dividen en elementos) para computar la estadística de los mismos dentro del conjunto principal. Es decir, teniendo los datos, estima con qué probabilidad los elementos del tercer nivel se conforman en el segundo, y el segundo en el primero.

Se aplica en casos de modelos de mezcla de pertenencias (mixed membership models), en el que cada dato puede pertenecer a diferentes clases de manera simultánea. Cada mezcla de temas está distribuida según la distribución de Dirichlet, $\text{Dir}(\alpha)$, con α positivo.

4.4.1 Terminología

Describimos los siguientes términos, con los que nos vamos a ir refiriendo a lo largo del proyecto:

- **Palabra**, es la unidad básica de los datos discretos. Es un ítem de un vocabulario que se indexa de manera que éste contiene todas las palabras $\{1, \dots, V\}$, donde una palabra se denota como un vector de dimensión V :

$$w = (0, \dots, 0, 1, 0, \dots, 0) \quad (9)$$

Por lo que:

$$\begin{aligned} w^u &= 0 \quad \forall u \neq v \\ w^u &= 1 \quad \text{si } u = v \end{aligned} \quad (10)$$

- **Tema**, siendo una secuencia de N palabras:

$$\bar{w} = (w_1, \dots, w_N) \quad (11)$$

- **Documento**, como conjunto de T temas :

$$D = (\bar{w}_1, \dots, \bar{w}_T) \quad (12)$$

Con esto no solamente buscamos modelar un conjunto que asigna probabilidades a los miembros en sí, sino que también se sugiere asignar la probabilidad de que otros documentos sean similares (Blei, Ng y Edu 2003).

Utilizando esta terminología, podemos dar un ejemplo de lo que pretende este algoritmo. Este proyecto se puede considerar un documento donde aparecen múltiples temas, siendo las palabras del mismo las que marcan de qué tema estamos hablando. Es decir, las palabras con se combinan de manera que forman un tema con cierta probabilidad, por lo tanto podemos considerar el documento una mezcla probabilística de estos temas.

4.4.2 Algoritmo y probabilidades

Si tenemos T temas, podemos describir la probabilidad de una palabra w_i en un documento como (Griffiths et al. 2004):

$$P(w_i) = \sum_{j=1}^T P(w_i | z_i = j) \times P(z_i = j) \quad (13)$$

donde z_i es una variable latente que indica el tema de donde se ha obtenido la palabra w_i , por lo que definimos:

- $P(w_i | z_i = j)$ como la probabilidad de escoger una palabra w_i en el tema z_i
- $P(z_i = j)$ es la probabilidad de coger una palabra del tema j en el documento en cuestión (que, evidentemente, variará según el documento elegido. Siguiendo con el ejemplo anterior, este proyecto y otro que trate un tema parecido podrá tener temas iguales con más o menos probabilidad y por tanto, palabras que conformen esos temas, también con una probabilidad determinada).

En este método consideramos los documentos como una mezcla de temas que aparecen con una probabilidad determinada. Esto nos permite descubrir el conjunto de temas utilizados en una colección de documentos.

Es decir, dados D documentos conteniendo T temas expresados en W palabras, presentamos:

- $P(w | z)$ como un conjunto de T distribuciones multinomiales ϕ sobre las W palabras, tal que $P(w | z = j) = \phi_q^{(j)}$.
- $P(z)$ con una serie de D distribuciones multinomiales θ sobre los T temas, por lo que para una palabra en el documento d describimos la probabilidad de manera que $P(z = j) = \theta_j^{(d)}$.

Una vez hecha esta parametrización, el algoritmo LDA es **generativo** de documentos completos, en el que se especifica un procedimiento probabilístico con los que nuevos documentos se generen con ciertos temas.

De esta manera, adopta el siguiente procedimiento generativo para cada tema en un documento (Blei, Ng y Edu 2003):

1. Escoge $N \sim \text{Poisson}(\xi)$.
2. Escoge $\theta \sim \text{Poisson}(\xi)$.
3. Para cada una de las N palabras w_n :
 - a. Se escoge un tema $z_n \sim \text{Multinomial}(\theta)$.
 - b. Se escoge una palabra w_n de $P(w_n | z_n, \beta)$, probabilidad multinomial condicionada al tema.

Se hacen una serie de suposiciones que simplifican el modelo, como que la dimensionalidad k de la distribución de Dirichlet y por tanto de la variable del tema es conocida y fija, y que la probabilidad de la palabra está parametrizada por una matriz $\beta \in k \times V$, donde:

$$\beta_{ij} = p(w^j = 1 \mid z^i = 1) \quad (14)$$

que simplemente tratamos como una cantidad fija que tiene que ser estimada.

Este modelo probabilístico lo podemos mostrar gráficamente en la Figura 22.

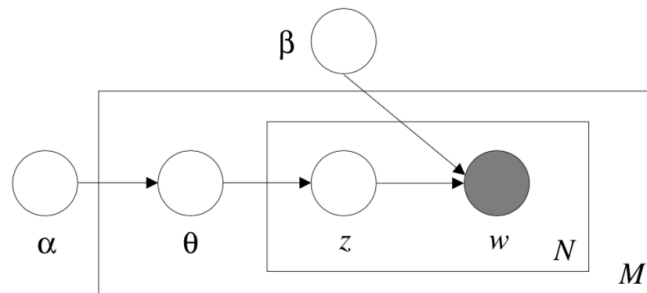


Figura 22. Algoritmo de generación LDA

Como vemos, este tiene tres niveles de representación. Los parámetros α y β son del nivel del documento (sin repetición), θ_d son del nivel de los temas (sin repetición) y z_{dn} y w_{dn} son variables de nivel de palabra (sin repetición).

4.5 Presentación de resultados

Una vez presentada la base de datos, la librería y el fichero de ejemplo, procedemos a presentar los resultados que hemos obtenido.

Primeramente, tras el tratado de los datos, el procesamiento y el entrenamiento de los algoritmos, los resultados de la predicción de las letras los mostramos en la Figura 23a y en la Figura 23b, correspondiendo con el sujeto A y el sujeto B.

- a -

Result for subject 1
Constructed labels: WQXPLZCOMRKOW7YFZDEZ1DPI9NN2GRKDJCUJRMEU0CJD2UFYP006J7LDGYEG0A5VHNEKBW4001TD0ILUEE5BFAEEXAW_K3R3MRU
True labels : WQXPLZCOMRK097YFZDEZ1DPI9NNVGRQDJCUVRMEU000JD2UFYP006J7LDGYEG0A5VHNEHBTX001TD0ILUEE5BFAEEXAW_K4R3MRU
Accuracy: 91.0%

- b -

Result for subject 2
Constructed labels: MERMIR00MUZJPXJOHUVFB0RZP3GL007AUFDKFTWE00ALZOP9R1CGZE11Y19EWX65QUYU7NAK_1ACJDVDNGQX0JB2EV2B5EFDIDTR
True labels : MERMIR00MUHJPXJOHUVLE0RZP3GL007AUFDKFTWE00ALZOP9R0CGZET1Y19EWX65QUYU7NAK_4YCDVDNGQX0DBEV2B5EFDIDNR
Accuracy: 91.0%

Figura 23. Predicción de las letras y comparación con la secuencia correcta

Como vemos, en ambas situaciones nos encontramos con una precisión del 91%. Deducimos que es un buen resultado para un sistema de ejemplo con un procesamiento relativamente simple, ya que a pesar de los algoritmos utilizados, el preprocesado es muy básico.

Las secuencias siguientes difieren de las originales:

Tabla 7. Comparación de resultados correctos y predichos, sujeto A

CARACTER ORIGINAL	CARACTER PREDICHO	COORDENADAS CORRECTAS	COORDENADAS PREDICHAS	REPETICIONES
W	9	(10, 5)	(12, 5)	1
2	V	(11, 4)	(10, 4)	1
K	Q	(8, 5)	(9, 5)	2
J	V	(8, 4)	(10, 4)	2
C	O	(7, 3)	(9, 3)	1
K	H	(8, 5)	(8, 2)	1
W	T	(10, 5)	(10, 2)	1
4	X	(11, 6)	(10, 6)	1
3	4	(11, 6)	(11, 5)	1

Tabla 8. Comparación de resultados correctos y predichos, sujeto B

CARACTER ORIGINAL	CARACTER PREDICHO	COORDENADAS CORRECTAS	COORDENADAS PREDICHAS	REPETICIONES
Z	H	(11, 2)	(8, 2)	1
F	L	(7, 6)	(8, 6)	1
B	E	(7, 2)	(7, 5)	1
1	O	(11, 3)	(9, 3)	1
1	T	(11, 3)	(10, 2)	1
1	4	(11, 3)	(11, 6)	1
A	Y	(7, 1)	(11, 1)	1
J	D	(8, 4)	(7, 4)	1
T	N	(10, 2)	(9, 2)	1

Observamos un cierto patrón:

- En 14 ocasiones de 20 (un 70 %) encontramos que el algoritmo está fallando en la predicción de las filas. Es decir, la columna que ofrece es la correcta, pero se mueve entre 1 y 4 filas.
- En 5 ocasiones de 20 (un 25 %) encontramos que el algoritmo está fallando en la predicción de las columnas. Es decir, la fila ofrecida es correcta pero se mueve entre 1 y 3 columnas.
- En 1 ocasión de 20 (un 5 %) encontramos que el algoritmo falla completamente. Es decir, no acierta ni fila ni columna.

Como vemos en la Figura 23, en ambos casos tenemos una precisión del 91 %, lo que nos hace generalizar sobre ello a una precisión del algoritmo de predicción en general del 91 %.

Lo siguiente que mostramos es la detección de la onda P3 en el electroencefalograma de ambos sujetos, en la Figura 24 a y b respectivamente.

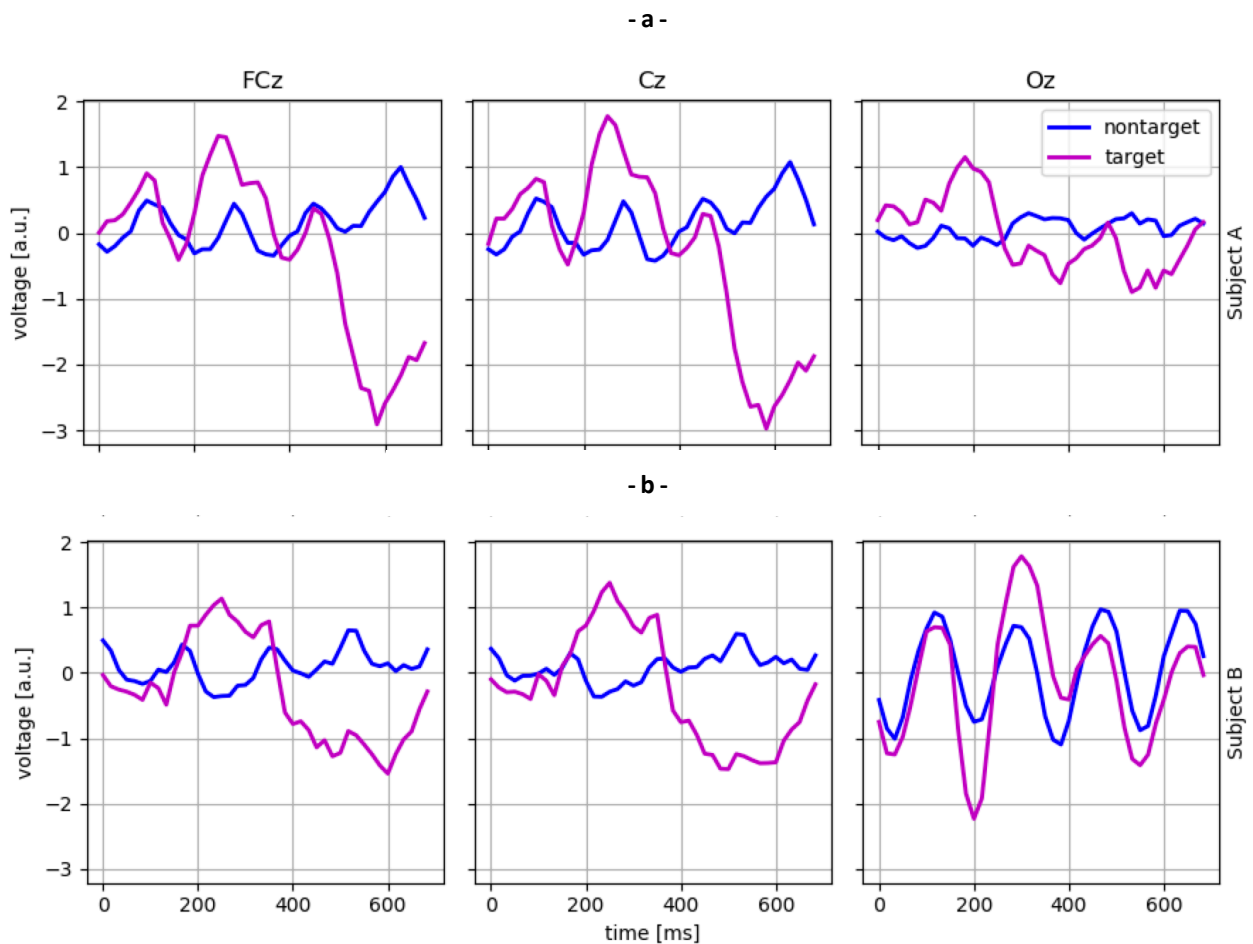


Figura 24. Presentación de la predicción de la onda P300 en los canales FCz, Cz y Oz, para el caso de aparición de target y no target, cada uno con su correspondiente leyenda

En esta gráfica podemos observar qué pasa con la detección de la onda P300 en cada uno de los sujetos. En el *eje y* tenemos los voltios normalizados que se presentan en el EEG, donde 0 es la referencia utilizada – como sabemos en el experimento el promedio de todos los canales. En el *eje x* tenemos los intervalos de tiempo donde se estudian y se genera la segmentación de los datos.

En azul observamos la respuesta cerebral tal cual, en general, cuando ningún target aparece en la pantalla. Es decir, es puro “ruido” de fondo que en general ronda siempre los mismos valores. En rosa, observamos la respuesta cuando aparece un target en la pantalla. La onda P300 ha sido detectada por completo. Sin embargo, hacemos un pequeño apunte:

- En los canales FCz y Cz se observa con gran claridad la onda, que aproximadamente obtiene su mayor pico 300 ms después del estímulo. Sin embargo, en el caso del sujeto A se ve con mayor claridad que en el sujeto B.
- En el canal Oz podemos ver que, aunque en el A se aprecia algo, en el B siquiera es perceptible a la vista en comparación con el EEG de fondo.

También vemos la diferencia entre el sujeto A y el sujeto B, siendo en ambos casos más amplia la señal en el primer sujeto.

Presentamos la distribución de detección de la onda en los sensores del EEG en la Figura 25 a y b, correspondientes a los sujetos A y B respectivamente.

Como vemos en la leyenda a la derecha de la Figura 25, se representa el voltaje en cada sensor que se presenta en el dibujo de la cabeza, siendo el mayor 2 y el menor -2.

En la primera fila, tendremos la representación del ruido de fondo del EEG, es decir, cuando en la pantalla no presenta ningún target. Observamos que está en torno al cero en ambos casos, poco más o poco menos, pero no tiene gran intensidad.

En la segunda fila, tendremos la presentación de cómo se va produciendo la onda P300 y su amplitud en cada época, desde que aparece el estímulo hasta aproximadamente 700 ms después del mismo. Como observamos, en ambos casos se repite que en el primer intervalo se comienza a ver cómo está creciendo una onda poco a poco, alcanzando su máximo en el segundo y tercer intervalo y llegando a valores muy negativos en el último intervalo estudiado. Es decir, con esto, tenemos el ciclo de vida de la onda P3 y de cómo se va distribuyendo temporal y espacialmente.

Se presentan también varios detalles sobre estos resultados:

- En el sujeto A, los niveles son más bajos en general que en el sujeto B, no solamente en la aparición del estímulo y por tanto del potencial evocado, si no en el EEG de fondo.
- Respecto al EEG de fondo no hay apenas diferencias más que la previamente dicha con respecto a la distribución de la misma.

- Respecto a la aparición del ERP, se ve una gran diferencia. En el sujeto A, comienza a aumentar el voltaje de los sensores Occipitales, distribuyéndose a prácticamente el resto del cerebro en el segundo intervalo de tiempo, comenzando a disminuir en el tercer intervalo y llegando al mínimo de voltaje en el último, alcanzando el pico más negativo. En el sujeto B se ve en los lóbulos Frontales y centrales un aumento del voltaje, así como una enorme disminución del mismo en el lóbulo Occipital. Poco a poco, va aumentando el voltaje en todo el área cerebral, hasta llegar a su pico máximo en el tercer intervalo de tiempo, sobretodo en el área Occipital. Por último, el voltaje decrece hasta puntos negativos, aunque no tanto como en el sujeto A.



Figura 25. Distribución en el EEG de la amplitud de generación de la onda P300, siguiendo una evolución temporal del estímulo desde 0 ms a 700 ms aproximadamente

Por último se representa r^2 en los diferentes canales centrales distribuidos en toda la cabeza, desde el Fpz hasta el Oz en la Figura 26.

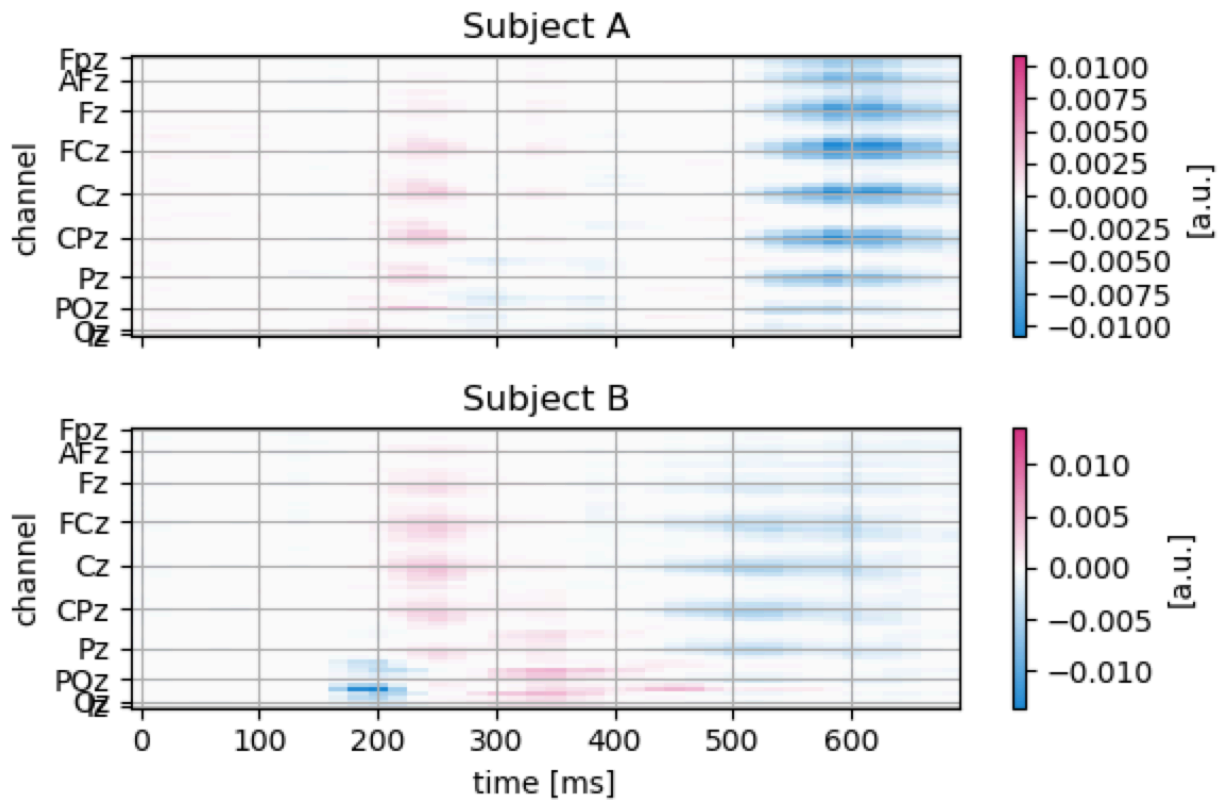


Figura 26. Representación del voltaje según los canales y épocas

Esto no es más que una representación en barras de lo que hemos visto antes distribuido en el cerebro, en la que la distribución es prácticamente la misma. Se hace evidente la presentación de un estímulo en todos los sensores hasta los 400 ms y una disminución como consecuencia hasta los 700 ms.

Apuntamos un último dato, viendo un pico de mayor voltaje en el sujeto B poco más tarde de los 300 ms y un pico de menor voltaje en el sujeto A cuando el estímulo desaparece por completo, alrededor de los 600 ms.

5 CONCLUSIONES

En el apartado anterior se mostró todo lo utilizado en este trabajo. Se explican todos los procedimientos seguidos a lo largo del experimento, así como los resultados que hemos ido obteniendo a medida que se ha ido ejecutando nuestro código de ejemplo, así como el uso de la librería que queríamos estudiar mediante la Base de Datos obtenida, pudiendo así completar un análisis de los mismos de manera más exhaustiva.

En este capítulo abordaremos el tema desde una perspectiva analítica, no solamente comparativa, en el que intentamos deducir qué conclusiones sacamos según estos resultados obtenidos, teorizando sobre las posibilidades de por qué ha ocurrido de esta manera y tratando de dar diversas soluciones.

5.1 Predicción de caracteres

Con los resultados que nos ha dado respecto a la predicción de letras, primeramente vemos en las figuras presentadas que no son malos resultados. Es decir, en porcentaje, obtenemos una precisión del 91 %, tanto para cada sujeto como generalización del algoritmo.

Como hemos visto, el algoritmo utilizado nos arroja unas predicciones que, aunque no del todo exactas y mejorables, en la mayoría de los casos únicamente se equivoca en la codificación de una de las coordenadas en la matriz. Esto quiere decir que tras el procesado de la señal y la ejecución del algoritmo LDA, obtenemos de manera aproximada la letra. Solamente en uno de los casos en los que el resultado no es correcto, el algoritmo estudiado se equivoca tanto en la fila como en la columna.

Una de las causas de estas equivocaciones en el mismo código puede ser la simpleza del preprocesado de la señal. Sabemos que en un electroencefalograma tenemos muchas señales que son, para este caso, ruido de fondo, por lo que es necesario eliminarlas de nuestra banda. Posiblemente, este preprocesado es bastante simple para lo que la predicción se refiere. Es probable que si mejorásemos este paso con un procesamiento de estas señales más elaborado y complejo, pudiésemos tomar una señal con menos ruido que nos permitiese una mejor detección de la onda P300.

Por otra parte, a pesar de que esto nos facilita bastante los siguientes procesos e incluso computacionalmente es mucho mejor, se hace un downsampling que produce una reducción en las muestras de nuestras señales. Esto también puede haber influido a la hora de analizar los resultados por parte del resto de nuestro fichero.

También podemos imputarlo al propio electroencefalograma, es decir, que el posicionamiento no sea del todo correcto, que los electrodos estén mal puestos y todos los factores que influyen a la hora de recoger un EEG. Por otra parte, sabemos que este método de obtención depende mucho del propio aparato, como los montajes posiblemente utilizados mal hechos (montajes unipolares, bipolares, etc).

Además de todo lo previamente descrito, que achaca todos los problemas al software de predicción, también podemos analizar desde una perspectiva más detallada nuestra Base de Datos. Así mismo, no tenemos mas información sobre la misma más que lo puramente dicho.

Este experimento tiene muchas variantes, así como muchos detalles y formas de ejecución que nos van a producir unos resultados u otros. Observándolo desde una perspectiva de un nivel más alto, el paradigma oddball, así como el del Speller (Farwell y Donchin 1988), tienen mucho margen de mejora.

Como vemos, es del año 1998, así como el experimento de 2004 y el software utilizado de 2014, por lo que lleva unos años de desventaja respecto a otros experimentos y bases de datos más actuales. Sin ir más lejos, la obtención de los potenciales evocados con el paradigma descrito se ha mejorado, así como se han estudiado diferentes factores que influirán en la amplitud de la onda. En el estudio realizado en 2010 (Citi, Poli y Cinel 2010), se propone una documentación teórica del modelo mejorado, haciendo una explotación de la fuente con variaciones de P300, variando el intervalo de tiempo entre estímulos.

Por otra parte, hemos explicado la importancia del tiempo de recuperación entre estímulos target. Sabemos que si dos estímulos target se presentan seguidos, el cerebro no es capaz de producir la onda P300 con tanta amplitud. Por tanto deducimos que para obtener mejores resultados se podría tener en cuenta este tiempo, de manera que consigamos unas ondas de mejor calidad y por tanto mejor predicción de las mismas. Tenemos dos soluciones posibles: introducir más estímulos no-target entre estímulos target o tener un intervalo entre intensificaciones mayor para que al cerebro le de tiempo recuperarse.

Sin embargo esto es complicado. Por un lado, desconocemos cuál es el estímulo target buscado, por lo que no es posible hacer a priori una secuencia de intensificaciones que separe lo suficiente los estímulos target. Por otro lado, esperar demasiado entre un estímulo y el siguiente (sea o no target), puede retrasar bastante el experimento, cuando el tiempo en este caso es algo esencial.

5.2 Detección de ondas en diferentes canales

En la representación gráfica de los tres canales centrales escogidos, podemos hacer diferentes apuntes.

Se nos presenta evidente la diferencia entre cada sujeto, por lo que dependiendo del mismo se arrojará un resultado u otro en diferentes sensores. A pesar de los estudios teóricos que indican que la onda P300 se estudia mejor en la zona parietal, se observa que la aparición del pico más alto varía.

Observamos que en el sujeto A los tres canales estudiados es visible la aparición del ERP. Sin embargo, en el sensor occipital la diferencia entre el EEG de fondo y la onda es nimia y puede producir errores en la predicción. En el sujeto B es bastante más complicado de observar a simple vista, ya que la diferencia entre el EEG de fondo cuando no hay estímulo y cuando lo hay, es bastante pequeña, e incluso en la zona occipital es prácticamente inidentificable.

Esto puede estar producido por la actividad cerebral del sujeto en cuestión, condiciones fisionómicas, grado de atención del mismo a la ejecución del experimento, propios factores personales, etc.

Como solución a este problema, se sugiere un estudio exhaustivo de todos los canales del EEG, que aunque con mayor carga computacional y quizás tiempo, sea más efectivo a la hora de hacer un algoritmo de predicción según el pico más alto.

5.3 Gráficas cerebrales

Al observar los resultados obtenidos en el experimento con respecto a la distribución de las ondas, podemos sacar conclusiones parecidas a las vistas en el apartado anterior. Se ha deseado hacer un estudio sobre la corteza cerebral del voltaje tanto del EEG de fondo como de lo que ocurre según las épocas y el espacio.

Hemos remarcado previamente las diferencias entre los sujetos A y B, aunque deseamos hacer una apreciación más exhaustiva sobre el tema.

Por una parte, volvemos a poner en el punto de mira las condiciones únicas de los sujetos en cuestión. Vemos que siquiera el EEG de fondo en ambos casos es igual, ya que se observa que en el sujeto B hay mucha más actividad neuronal de fondo que en el sujeto A.

Por otra parte, en la aparición del estímulo, se distribuye de diferente manera no solamente en los sensores en sí si no en el tiempo que tarda en producirse.

Esto ya lo hemos comentado antes, haciéndose evidente de forma experimental que cada persona requiere un tiempo de asimilación y procesamiento del estímulo. De hecho, ya sabíamos previamente que la onda P3 se clasifica como la tercera onda positiva que aparece por diversas razones y que su aparición puede alargarse o adelantarse en el tiempo entre los 250 y 700 ms tras el estímulo.

De esta manera, tendremos que tener en cuenta la dependencia del experimento con el sujeto.

En la siguiente gráfica que se muestra de r^2 nos ofrece los mismos resultados, por lo que no queda por hacer más reseña sobre la distribución espacio-temporal de la onda P300.

Con respecto a esto, proponemos como solución un software más efectivo, en el que se haga un aprendizaje intensivo previo a la instalación de un sistema como este. Es decir, que la propia BCI sea capaz de aprender cuáles son las distribuciones y los tiempos que aparecen en ese sujeto, así haciendo un modelo estadístico en base a este aprendizaje, y siendo capaz de reafirmarse y optimizarlo en el caso de ser necesario.

De la misma forma, también se hace necesario tener unos límites de error sobre la predicción de caracteres, por lo que es importante un feedback inicial por parte del sujeto.

5.4 Generalidades

Como conclusiones a trazas más generales podemos resumir este estudio de diferentes maneras:

- Asegurarse de un porcentaje de precisión en la predicción de los algoritmos, haciendo un sistema retroalimentativo software-sujeto para una implantación óptima.
- Una actualización de los sistemas, no únicamente físicos como nuevos equipos de captación del EEG, si no informáticos, así como un estudio exhaustivo de nuevas plataformas que proporcionan una mejoría en la detección de ERPs.

6 LÍNEAS FUTURAS

Se hace evidente la importancia de estos estudios. El cerebro es el gran desconocido de nuestro cuerpo. Pese a todos los años que han pasado y la cantidad de estudios realizados sobre el tema, sigue siendo fruto de curiosidad por parte de toda la unidad neurocientífica, bioingeniera, psiquiátrica, entre otras muchas.

Sabemos que en nuestro cerebro se produce absolutamente todo lo que ocurre en nuestro cuerpo, pero no sabemos identificar mas que una serie de ondas eléctricas que están ahí. El tema de investigación que se ha tratado en este proyecto aún siguen siendo simples pruebas que necesitan un gran avance.

De esta manera, trataremos en este punto la situación actual del análisis de señales electroencefalográficas, así como las últimas investigaciones sobre el tema.

6.1 Investigación actual

Durante todo este proyecto, se ha tratado de dar unas bases teóricas para llegar a experimentar y documentar un experimento de gran utilidad que empezó en el 1998. El Donchin Speller basado en la onda P300 no se ha quedado ni mucho menos obsoleto, si no que es un motivo de investigación muy potente actualmente debido a su gran potencial.

Se plantea como factible la implantación de este modelo y de diferentes BCI sobretudo en pacientes con discapacidad, ya que esta herramienta puede proporcionar una comunicación que no requiera actividad muscular.

Aquí surge un problema, ya que la mayoría de estudios realizados tienen una tasa muy baja de predicción de letras, lo que limita una comunicación fluida. Hablamos de únicamente unos pocos caracteres por minuto, cuando en una conversación normal se forman palabras completas en cuestión de segundos.

6.1.1 Speller mediante un ECoG

Como hemos dicho, es muy importante la rapidez de traducción cerebro-computadora. Uno de los motivos por los que esto carece de velocidad puede ser las limitaciones de un electroencefalograma superficial, por lo que se plantea como una posible solución una BCI que actúe directamente con ondas captadas mediante sensores colocados quirúrgicamente sobre el cerebro expuesto.

Se propone de esta manera un speller que utilice diferentes ERPs, entre ellos la P300, en el que el sujeto piense en un carácter en una matriz mientras ésta va intensificado sus columnas y filas muy rápidamente y de manera pseudo-aleatoria. Esto, con todo lo estudiado previamente, sabemos que producirá una respuesta en nuestro cerebro cuando coincida la intensificación con el carácter en el que el sujeto piensa.

Con este experimento (Brunner et al. 2011) se llega a la conclusión de que las señales que provee el ECoG puede predecir caracteres en una BCI basada en el Donchin Speller con una tasa media de 17 caracteres por minuto (69 bits/min) y una tasa pico de 22 caracteres por minuto (113 bits/min). Con respecto a estudios anteriores, se deduce que es una tasa aproximadamente 3 o 4 veces mayor que las previamente estudiadas.

6.1.2 Nuevos conceptos del Speller

Como hemos descrito antes, una de las aplicaciones más importantes de este experimento es la adaptación de estos sistemas en personas con discapacidad que no sean capaces de comunicarse normalmente. Aquí surge un nuevo problema, ya que el Donchin Speller tradicional descrito limita su aplicación a personas con una visión correcta. Se estudian nuevas soluciones, como sistemas de deletreo que no necesiten estimulación visual, pero la mayoría tienen unos resultados bastante pobres.

Sin embargo, se pone en línea de investigación una extensión de la matriz original, en la que no solamente se utiliza la interfaz para estímulos visuales, si no que se pueda compaginar con únicamente estímulos auditivos o ambos a la vez. De esta manera, el sistema en cuestión será capaz de adaptarse a las necesidades del sujeto, pudiendo elegir cuál de las opciones es mejor para el mismo (Belitski, Farquhar y Desain 2011). La actuación de este experimento más baja que los que utilizan el estímulo visual, pero con resultados bastante mejores que otras BCI basados en estímulos auditivos.

A trazos generales, la propuesta es, por una parte, un aprendizaje sobre cómo hacer un “paradigma oddball” mediante estímulos auditivos y cuál ofrecía mejores resultados y por otra parte el estudio de estímulos propiamente auditivos, visuales y audio-visuales.

Resulta interesante la metodología utilizada, ya que para mejorar la actuación de los estímulos auditivos, se propone asignar un sonido a cada letra de la matriz. Se muestra en la Figura 27 lo hecho:

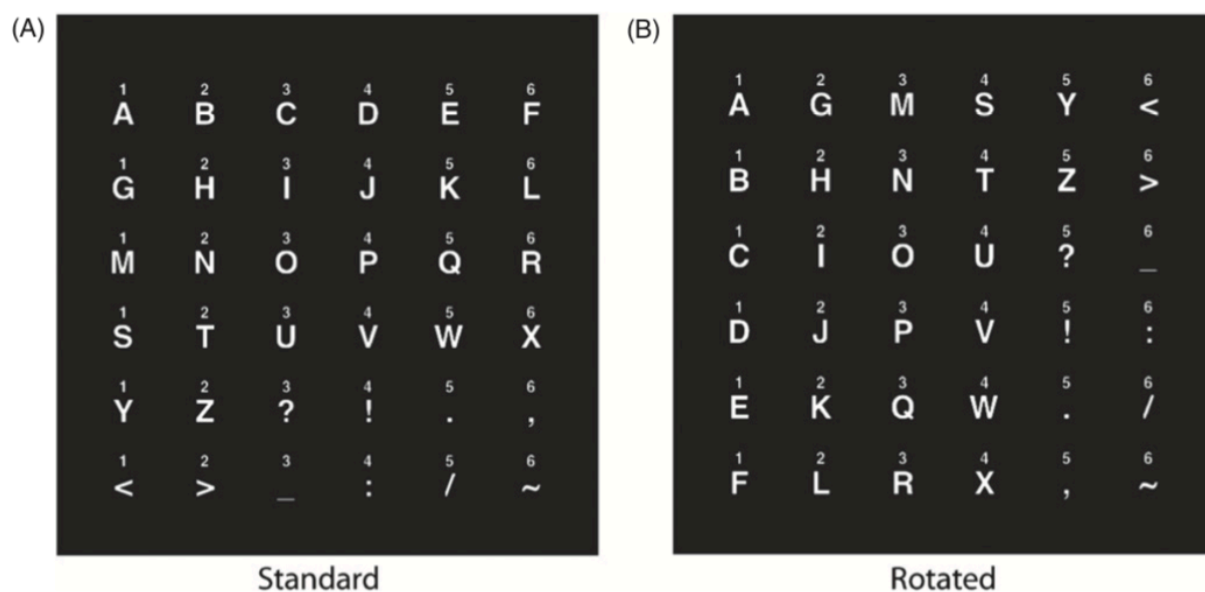


Figura 27. Mejora del Donchin Speller tradicional a un sistema audio-visual

Primeramente se muestra la matriz tradicional, a la que se le entrena con un sonido, y posteriormente se rota la matriz 90° y se vuelven a reproducir los sonidos, para que el sujeto interiorice con más facilidad la asignación carácter-sonido.

Nos resulta interesante recalcar no solamente la adaptación a una comunicación normal a un sector de pacientes, si no intentar generalizar lo máximo posible para que esto sea efectivo para absolutamente todas las personas.

También se propone un nuevo método de aprendizaje para la mejora de los algoritmos de la propia BCI, de manera que se le añade al experimento cuatro opciones para responder preguntas de sí/no (Sellers, Kübler y Donchin 2006). Se le presenta al usuario una secuencia de cuatro eventos, que son las palabras “sí”, “no”, “pasar” y “terminar”, utilizando también estímulos audio-visuales.

6.1.3 Estudio de diferentes algoritmos de diferenciación

Otra de las investigaciones respecto a este tema se basa en la mejora del propio algoritmo de detección y el sistema de captación de la BCI.

Se proponen y se comparan muchas técnicas de clasificación de la onda P300 (Krusienski et al. 2006), como los métodos PCM (Pearson's Correlation Method), FLD (Fisher's Linear Discriminant), SWLDA (Stepwise Linear Discriminant Analysis), LSVM (Linear Support Vector Machine) o GSVM (Gaussian Support Vector Machine).

Esto da lugar a diferentes métodos de detección de la presencia o no presencia de la onda P300 en un EEG, que son actualmente objetos de estudio para la mejora de la tasa de predicción o la tasa de errores entre otras.

Por otro lado se propone cómo afecta el tamaño de la matriz de caracteres a la hora de la aparición de la onda P300, así como el intervalo de tiempo entre intensificaciones (ISI) (Sellers, Kübler y Donchin 2006). Se reducen las dimensiones a matrices de 3 x 3, además de la original de 6 x 6.

Para el experimento online, la precisión es mucho mayor para las matrices de 3x3 con un ISI = 175 ms, aunque la tasa de bit se propone mayor para las matriz de 6 x 6 con el mismo ISI. Se muestra en la Figura 28 las matrices utilizadas en el ejemplo.

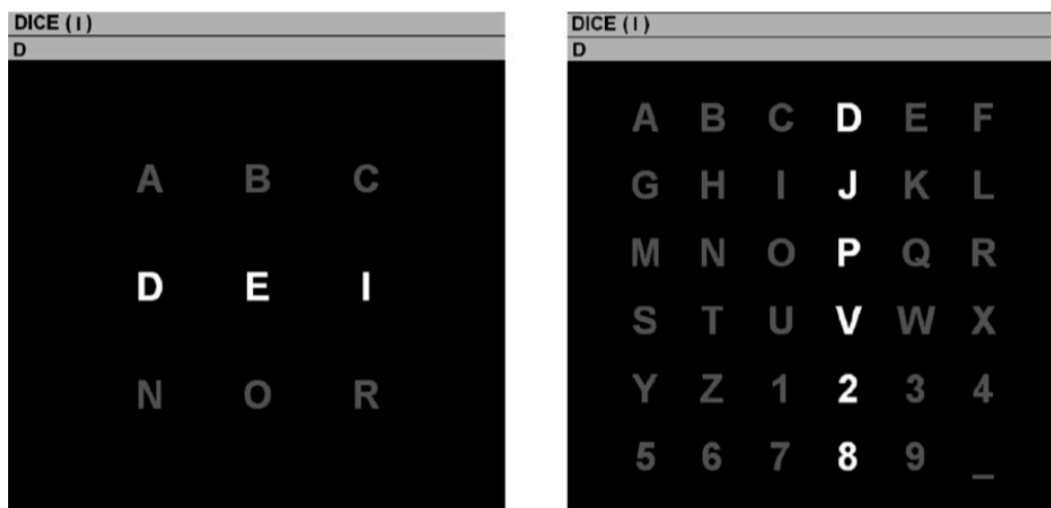


Figura 28. Nuevas matrices propuestas para nuevos estudios del Donchin Speller

6.2 Líneas futuras en base a las conclusiones obtenidas

Podemos proponer una serie de líneas de investigación con respecto a las conclusiones y resultados propuestos en este Trabajo de Fin de Grado:

- Respecto a la precisión a la hora de adivinar la letra en cuestión tras el experimento, sabemos que los diferentes procesados de señal y algoritmos utilizados nos darán unos resultados bastante diferentes, por lo que resultaría interesante un estudio intensivo de nuevos algoritmos de predicción, así como de clasificación de datos. También es importante tener en cuenta que el procesado de la señal del EEG es muy relevante a la hora de predecir la onda P300 en ellas, por lo que resulta evidente que un tratamiento previo exhaustivo de la señal probablemente mejore el funcionamiento de la predicción. Cuanto más limpia tengamos nuestras señales, mejores resultados podremos obtener.
- Se presenta muy interesante la proposición de un estudio intensivo de todas los canales del electroencefalograma, ya que la distribución de la onda en cada sujeto es diferente. A pesar de saber que hay ciertos sensores en los que esta onda se produce con mayor intensidad, se puede optimizar la predicción si se caracterizan todos los canales utilizados, y se propone una comparación mediante el análisis de esos datos para un estudio estadístico de cuál es la letra original con unos márgenes de precisión.
- Creemos también importante que un aprendizaje sobre el propio sujeto previo a los experimentos puede ser muy productivo, ya que al tener tanta diferencia entre los sujetos, podemos retroalimentar nuestra BCI con esa información para optimizar su actuación.
- Proponemos también que sería muy productivo estudiar los factores que puedan afectar a la aparición de la onda P300, por lo que optimizar lo que es el experimento en sí variando estos factores también parece algo factible y de gran utilidad.

7 BIBLIOGRAFÍA

- ANACONDA INC, 2019. *Anaconda Python/R Distribution - Anaconda* [en línea]. 2019. S.l.: s.n. Disponible en: <https://www.anaconda.com/distribution/>.
- ARBOLEDA GÓMEZ, J.E., 2011. Implementación Técnica De La Ubicación De Focos Epileptogénicos Mediante El Uso De Imágenes Multimodales (Siscom). , pp. 67.
- BAILEY, R., 2017. Learn About Temporal Lobes in the Cerebral Cortex. [en línea]. Disponible en: <https://www.thoughtco.com/temporal-lobes-anatomy-373228>.
- BAILEY, R., 2018a. Divisions of the Brain. [en línea]. Disponible en: <https://www.thoughtco.com/divisions-of-the-brain-4032899>.
- BAILEY, R., 2018b. Frontal Lobes: Motor Cortex, Cognition, and Speech. [en línea]. Disponible en: <https://www.thoughtco.com/frontal-lobes-anatomy-373213>.
- BAILEY, R., 2018c. Occipital Lobes in the Cerebral Cortex. [en línea]. Disponible en: <https://www.thoughtco.com/occipital-lobes-anatomy-373224>.
- BAILEY, R., 2018d. Parietal Lobes Function and Brain Anatomy. [en línea]. Disponible en: <https://www.thoughtco.com/parietal-lobes-of-the-brain-3865903>.
- BAREA NAVARRO, R., [sin fecha]. Electroencefalografía. *Instrumentación Biomédica*. S.l.: Departamento Electrónica, Universidad Alcalá, pp. 26.
- BELITSKI, A., FARQUHAR, J. y DESAIN, P., 2011. P300 audio-visual speller. *Journal of Neural Engineering*. S.l.: s.n., ISBN 1741-2560. DOI 10.1088/1741-2560/8/2/025022.
- BLANKERTZ, B. y SCHALK, G., 2004. Documentation Wadsworth BCI Dataset (P300 Evoked Potentials) Data Acquired Using BCI2000's P3 Speller Paradigm BCI Competition III Challenge 2004. [en línea]. S.l.: Disponible en: <http://www.bci2000.org>.
- BLEI, D.M., NG, A.Y. y EDU, J.B., 2003. Latent Dirichlet Allocation Michael I. Jordan. *Journal of Machine Learning Research*. S.l.:

- BRUNNER, P., RITACCIO, A.L., EMRICH, J.F., BISCHOF, H. y SCHALK, G., 2011. Rapid communication with a «P300» matrix speller using electrocorticographic signals (ecog). *Frontiers in Neuroscience*, ISSN 16624548. DOI 10.3389/fnins.2011.00005.
- CAMPOS DOMÍNGUEZ, G., [sin fecha]. *ERPs*. S.l.: s.n.
- CHORNET, A. y OTEO, J.Á., 2002. Actividad eléctrica cerebral - Revista Mètode. [en línea]. Disponible en: <https://metode.es/revistas-metode/monograficos/actividad-electrica-cerebral-un-lenguaje-por-descifrar.html>.
- CITI, L., POLI, R. y CINEL, C., 2010. Documenting, modelling and exploiting P300 amplitude changes due to variable target delays in Donchin's speller. *Journal of Neural Engineering*, ISSN 17412560. DOI 10.1088/1741-2560/7/5/056006.
- FABIANI, M., GRATTON, G. y KARIS, D., 1987. *Reliability of Measurement of the P300 Component of the Event-Related Brain Potential*. S.l.: s.n. ISBN 0-89232-644-1.
- FARWELL, L.A. y DONCHIN, E., 1988. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, ISSN 00134694. DOI 10.1016/0013-4694(88)90149-6.
- GRIFFITHS, T.L., STEYVERS, M., BLEI, by y BLEI, J., 2004. Finding scientific topics A first step in identifying the content of a document is determining which topics that document addresses. We describe a generative model for documents, introduced. [en línea]. S.l.: Disponible en: www.pnas.org/cgi/doi/10.1073/pnas.0307752101.
- KHAN ACADEMY, [sin fecha]. La sinapsis. [en línea]. Disponible en: <https://es.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/the-synapse>.
- KOLDA, T.G. y BADER, B.W., 2009. Tensor Decompositions and Applications. *SIAM Review*, ISSN 0036-1445. DOI 10.1137/07070111X.
- KROPOTOV, J.D., 2009. Quantitative EEG, Event-Related Potentials and Neurotherapy. *Quantitative EEG, Event-Related Potentials and Neurotherapy*. S.l.: s.n., ISBN 9780123745125.
- KRUSIENSKI, D.J., SELLERS, E.W., CABESTAING, F., BAYOUDH, S., MCFARLAND, D.J., VAUGHAN, T.M. y WOLPAW, J.R., 2006. A comparison of classification techniques for the P300 Speller. *Journal of Neural Engineering*, ISSN 17412560. DOI 10.1088/1741-2560/3/4/007.
- LDT, T.C.T., 2012. 10 / 20 System Positioning. *Gloucester Road*, no. 2410, pp. 77-79.
- LEI, X. y LIAO, K., 2017. Understanding the influences of EEG reference: A large-scale brain network perspective. *Frontiers in Neuroscience*, vol. 11, no. APR, pp. 1-11. ISSN 1662453X. DOI 10.3389/fnins.2017.00205.

- LINDEN, D.E.J., 2005. *The P300: Where in the brain is it produced and what does it tell us?* 2005. S.l.: s.n. ISBN 1073858405.
- MACHADO, S., ARAÚJO, F., PAES, F., VELASQUES, B., CUNHA, M., BUDDE, H., BASILE, L.F., ANGHINAH, R., ARIAS-CARRIÓN, O., CAGY, M., PIEDADE, R., DE GRAAF, T.A., SACK, A.T. y RIBEIRO, P., 2010. *EEG-based brain-computer interfaces: An overview of basic concepts and clinical applications in neurorehabilitation*. 2010. S.l.: s.n. ISBN 0334-1763 (Print)r0334-1763 (Linking).
- MCFARLAND, D.J. y WOLPAW, J.R., 2017. EEG-based brain-computer interfaces. *Current Opinion in Biomedical Engineering* [en línea], vol. 4, pp. 194-200. ISSN 01681176. DOI 10.1016/0168-1176(84)80033-6. Disponible en: <http://linkinghub.elsevier.com/retrieve/pii/S246845111730082X>.
- NCAN, [sin fecha]. BCI2000 | Schalk Lab. [en línea]. Disponible en: <http://www.schalklab.org/research/bci2000>.
- NICOLAS-ALONSO, L.F. y GOMEZ-GIL, J., 2012. *Brain computer interfaces, a review*. 2012. S.l.: s.n. ISBN 0308018032250.
- NÚÑEZ-PEÑA, M.I., CORRAL, M.J. y ESCERRA, C., 2004. Potenciales evocados cerebrales en el contexto de la investigación psicológica: una actualización. *Anuario de Psicología*, vol. 35, pp. 3-21.
- OPENSTAX, 2013. Communication Between Neurons - Anatomy & Physiology - OpenStax CNX. 6 March [en línea]. Disponible en: <https://cnx.org/contents/FPtK1znh@6.27:p74vr6PZ@3/Communication-Between-Neurons>.
- ORTEGA, M., 2014. Electroencefalograma: Interpretación para mortales. [en línea]. Disponible en: <https://sapiensmedicus.org/electroencefalograma-interpretacion/>.
- PICTON, T.W., 1992. The P300 wave of the human event-related potential. *Journal of Clinical Neurophysiology*, ISSN 15371603. DOI 10.1097/00004691-199210000-00002.
- POLICH, J. y KOK, A., 1995. Cognitive and biological determinants of P300: an integrative review. *Biological Psychology*, ISSN 03010511. DOI 10.1016/0301-0511(95)05130-9.
- POLICH, J. y MARGALA, C., 1997. P300 and probability: Comparison of oddball and single-stimulus paradigms. *International Journal of Psychophysiology*, ISSN 01678760. DOI 10.1016/S0167-8760(96)00742-8.
- RAMOS-ARGÜELLES, F., MORALES, G., EGOZCUE, S., PABÓN, R.M. y ALONSO, M.T., 2009. Técnicas básicas de electroencefalografía: principios y aplicaciones clínicas. *Anales del sistema sanitario de Navarra*, vol. 32 Suppl 3, pp. 69-82. ISSN 11376627. DOI 10.4321/S1137-66272009000600006.
- SACEDA CORRALO, D., 2018. Electroencefalograma (EEG), qué es - Pruebas Médicas. [en línea]. Disponible en: <https://www.webconsultas.com/pruebas-medicas/electroencefalograma-eeg-12529>.

- SADDIQUE, S.M. y SIDDIQUI, L.H., 2009. EEG based brain computer interface. *Journal of Software*, ISSN 1796217X. DOI 10.4304/jsw.4.6.550-554.
- SANCHEZ SEISDEDOS, P., 2017. Funciones de las neuronas: Conoce el funcionamiento de estas células. 8 de agosto [en línea]. Disponible en: <https://blog.cognifit.com/es/funciones-de-las-neuronas/>.
- SANMILLÁN VIDOY, F.L., 2017. *Detección de Crisis Epilépticas a partir del Electroencefalograma*. S.l.: Universidad de Sevilla.
- SCHALK, G., DEAN, K. y WOLPAW, J.R., 2004. BCI Competition III. [en línea]. Disponible en: <http://www.bbci.de/competition/iii/>.
- SELLERS, E.W., KÜBLER, A. y DONCHIN, E., 2006. Brain-computer interface research at the University of South Florida cognitive psychophysiology laboratory: The P300 speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. S.l.: s.n., ISBN 5184747958. DOI 10.1109/TNSRE.2006.875580.
- SOLTANI, M. y KNIGHT, R.T., 2000. Neural Origins of the P300. *Critical ReviewsTM in Neurobiology*, ISSN 0892-0915. DOI 10.1615/CritRevNeurobiol.v14.i3-4.20.
- SOMANI, S. y SHUKLA, J., 2012. *The P300 wave of Event-Related-Potential*. S.l.: s.n.
- SÖRNMO, L. y LAGUNA, P., 2005. *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. S.l.: s.n. ISBN 9780124375529.
- SPYDER DOC CONTRIBUTORS, 2018. *Spyder* [en línea]. 2018. S.l.: s.n. Disponible en: <https://www.spyder-ide.org/>.
- ST. LOUIS, E., FREY, L., BRITTON, J., HOPP, J., KORB, P., KOUBEISSI, M., LIEVENS, W. y PESTANA-KNIGHT, E., 2016. *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants* [en línea]. S.l.: s.n. ISBN 9780997975604. Disponible en: <https://www.ncbi.nlm.nih.gov/books/NBK390354/>.
- TALAMILLO GARCÍA, T., 2011. Nociones elementales para la interpretación del EEG. . S.l.:
- TROUT, G. y MARTÍNEZ, M., 2006. Conceptos básicos de electroencefalografía. *Duazary*, vol. 3, pp. 18-23.
- VAID, S., SINGH, P. y KAUR, C., 2015. EEG signal analysis for BCI interface: A review. *International Conference on Advanced Computing and Communication Technologies, ACCT*. S.l.: s.n., ISBN 9781479984886. DOI 10.1109/ACCT.2015.72.

- VENTHUR, B., DÄHNE, S., HÖHNE, J., HELLER, H. y BLANKERTZ, B., 2015. Wyrn: A Brain-Computer Interface Toolbox in Python. *Neuroinformatics* [en línea], vol. 13, no. 4, pp. 471-486. ISSN 1539-2791. DOI 10.1007/s12021-015-9271-8. Disponible en: <http://link.springer.com/10.1007/s12021-015-9271-8>.
- YEUNG, N., BOGACZ, R., HOLROYD, C.B. y COHEN, J.D., 2004. Detection of synchronized oscillations in the electroencephalogram: An evaluation of methods. *Psychophysiology*, ISSN 00485772. DOI 10.1111/j.1469-8986.2004.00239.x.

ANEXO A: CÓDIGOS UTILIZADOS

speller.py

```
# coding: utf-8

# ### Getting the Dataset
#
# This example uses the [Data Set 2][bcicomp3ds2] from the BCI
# Competition 3.
# After downloading and copying it into a directory called `data` next
# to this script, you should be able to follow this example.
#
# [bcicomp3ds2]: http://www.bbc.de/competition/iii/#data\_set\_ii
#

# Todas las importaciones necesarias para el código

from __future__ import division

import numpy as np
import scipy as sp
from matplotlib import pyplot as plt
from matplotlib import ticker
import matplotlib as mpl

from wurm import plot
plot.beautify()
from wurm.types import Data
from wurm import processing as proc
from wurm.io import load_bcicomp3_ds2

# Se definen las variables que contendran los Training y de los Test
# reales de los dos sujetos en formato ".mat".
TRAIN_A = '../BCI_BBDD/Subject_A_Train.mat'
TRAIN_B = '../BCI_BBDD/Subject_B_Train.mat'

TEST_A = '../BCI_BBDD/Subject_A_Test.mat'
TEST_B = '../BCI_BBDD/Subject_B_Test.mat'

# El feedback dado por cada uno de los sujetos sobre los caracteres
# pensados
TRUE_LABELS_A =
    'WQXPLZCOMRKO97YFZDEZ1DPI9NNVGRQDJCUVRMEUOOOJD2UFYPOO6J7LDGY
    EGOA5VHNEHBTXOO1TDOILUEE5BFAEEXAW_K4R3MRU'
TRUE_LABELS_B =
```

```

'MERMIROOMUHJPXJOHUVLEORZP3GLOO7AUFDKFTWEOOALZOP9ROCGZET1Y1
9EWX65QUYU7NAK_4YCDVDNGQXODBEV2B5EFDIDNR'

# Matriz del speller (la que se vería en la pantalla LCD)
MATRIX = ['abcdef',
          'ghijkl',
          'mnopqr',
          'stuvwx',
          'yz1234',
          '56789_']

# Marcadores de target/no target o de fila/columna flasheada
MARKER_DEF_TRAIN = {'target': ['target'], 'nontarget': ['nontarget']}
MARKER_DEF_TEST = {'flashing': ['flashing']}

# El intervalo de tiempo de cada epoch
SEG_IVAL = [0, 700]

# Intervalos para hacer la media de los canales (de 150 a 220, de 200
# a 260...)
JUMPING_MEANS_IVALS_A = [150, 220], [200, 260], [310, 360], [550, 660]
# 91%
JUMPING_MEANS_IVALS_B = [150, 250], [200, 280], [280, 380], [480, 610]
# 91%

def preprocessing_simple(dat, MRK_DEF, *args, **kwargs):
    """ Simple preprocessing that reaches 97% accuracy. """

    # Toma la frecuencia tomada de los datos para posteriormente
    # meterla
    # en el filtro de Butterworth . F de Nyquist
    fs_n = dat.fs / 2

    # b, a = num, den del filtro de Butterworth
    # argumentos de entrada: orden, frecuencia importante, tipo de
    # filtro
    # Estamos utilizando un filtro paso de baja, donde la frecuencia
    # de corte es 10, y fs_n la frecuencia de Nyquist
    b, a = proc.signal.butter(5, [10 / fs_n], btype='low')
    # Filtra dos veces la señal con el filtro de Butterworth previamente
    # hecho
    dat = proc.filtfilt(dat, b, a)

    # Reduce la tasa de muestreo de la señal original (240 Hz) a
    # frecuencia
    # indicada en el parámetro, tomando de la señal de n en n muestras,
    # siendo n = fs/fssubsampling, por lo que fssubsampling debe ser un
    # múltiplo entero de la frecuencia de muestreo
    dat = proc.subsample(dat, 20)
    # Hace épocas de los datos según MRK_DEF pasado (si son los datos
    # de training o si son los de testeo), segmentándolos según los
    # intervalos dados por SEG_IVAL
    epo = proc.segment_dat(dat, MRK_DEF, SEG_IVAL)
    # Crea los vectores característicos de cada época. Objetos datos
    # en 2 dimensiones

```

```

fv = proc.create_feature_vectors(epo)
return fv, epo

def preprocessing(dat, MRK_DEF, JUMPING_MEANS_IVALS):
    # Ordenacion de los canales segun el sistema 10-20.
    # Orden: Frontal, Temporal, Central, Parietal, Occipital.
    #         por línea de sensores, de arriba hacia abajo,
    #         por cada línea, de hemisferio izquierdo al derecho
    dat = proc.sort_channels(dat)

    # Toma la frecuencia tomada de los datos para posteriormente
    # meterla en el filtro de Butterworth
    fs_n = dat.fs / 2 # Freq de Nyquist
    # Filtra la señal primero con un filtro LP y luego con uno HP
    b, a = proc.signal.butter(5, [30 / fs_n], btype='low')
    dat = proc.lfilter(dat, b, a)
    b, a = proc.signal.butter(5, [.4 / fs_n], btype='high')
    dat = proc.lfilter(dat, b, a)

    # Reduce la tasa de muestreo de la señal original (240 Hz) a la
    # frecuencia indicada en el parámetro, tomando de la señal de n en
    # n muestras, siendo n = fs/fssubsampling, por lo que fssubsampling
    # debe ser un múltiplo entero de la frecuencia de muestreo
    dat = proc.subsample(dat, 60)
    # Hace épocas de los datos según MRK_DEF pasado (si son los datos
    # de training o si son los de testeo), segmentándolos según los
    # intervalos dados por SEG_IVAL
    epo = proc.segment_dat(dat, MRK_DEF, SEG_IVAL)

    # Calcula las medias de los datos en epocas, segun los intervalos
    # dados saltos dados en la variable
    fv = proc.jumping_means(epo, JUMPING_MEANS_IVALS)
    # Crea los vectores característicos de cada epoca. Objetos dados
    # en 2 dimensiones
    fv = proc.create_feature_vectors(fv)
    return fv, epo

# Hace un bucle para asignar a las variables de control de sujetos y
# datos los valores adecuados según el sujeto que se contemple. Sujeto
# va a ser 0 y 1, correspondiendo al A y al B, definidas variables
# previamente en el código (generalmente estas variables son rutas que
# serán cargadas a posteriori en el código)
epo = [None, None]
acc = 0
for subject in range(2):
    if subject == 0: # sujeto A
        training_set = TRAIN_A # Valores de training de A
        testing_set = TEST_A # Valores de testing de A
        labels = TRUE_LABELS_A # Caracteres reales pensados por A
        jumping_means_ivals = JUMPING_MEANS_IVALS_A # Intervalos de
                                                    media A
    else: # sujeto B
        training_set = TRAIN_B # Valores de training de B
        testing_set = TEST_B # Valores de testing de B
        labels = TRUE_LABELS_B # Caracteres reales pensados por B

```

```

jumping_means_ivals = JUMPING_MEANS_IVALS_B # Intervalos de
                                              media B

# COMIENZA A TRATAR LOS DATOS SEGÚN LO CARGADO PREVIAMENTE
# load the training set. Convierte el training_set (variable que
# contiene la ruta de los datos del training) en el formato "data"
# de la librería para poder tratarlo posteriormente.
dat = load_bcicomp3_ds2(training_set)
# Función "preprocessing" descrita más arriba
fv_train, epo[subject] = preprocessing(dat, MARKER_DEF_TRAIN,
                                       jumping_means_ivals)

# train the lda (Algoritmo de discriminacion LDA). Recibe un vector
# bidimensional, que hemos sacado previamente en el preprocesado
# de las señales
cfy = proc.lda_train(fv_train)

# Carga el testing_set y preprocesado del mismo
dat = load_bcicomp3_ds2(testing_set)
fv_test, _ = preprocessing(dat, MARKER_DEF_TEST,
                           jumping_means_ivals)

# Prediccion segun el algoritmo LDA de los vectores caracteristicos
# desde los datos de testeo, previamente habiendolo entrenado
# (training)
lda_out_prob = proc.lda_apply(fv_test, cfy)

# Orden del del estimulo.

# Con el fv_test.stimulus_code, lo ordena en 100 matrices de 15x12,
# (sabemos que se hacen 15 intensificaciones en bloques de 12)
# haciendo con el argsort() que se ordenen los indices de mayor a
# menor segun los valores del stimulus_code
unscramble_idx = fv_test.stimulus_code.reshape(100, 15,
                                              12).argsort()

# Crea los array de (100,15,12) de los indices para poder acceder
# a ellos
static_idx = np.indices(unscramble_idx.shape)
# Crea el mismo array con igual orden despues de haber pasado el
# algoritmo LDA
lda_out_prob = lda_out_prob.reshape(100, 15, 12)
# Una vez lo ha ordenado, toma los datos que le interesa
lda_out_prob = lda_out_prob[static_idx[0], static_idx[1],
                           unscramble_idx]

# lda_out_prob = lda_out_prob[:, :5, :]

# destil the result of the 15 runs
lda_out_prob = lda_out_prob.sum(axis=1)

# Ordena las matrices de lda_out_prob, dejando en ella LOS INDICES
# en orden
lda_out_prob = lda_out_prob.argsort()

```

```

# Ordenamos las columnas y las filas que vamos a corresponder en
# la matriz según si lda_out_prob sea mayor o menor que 5. Obtenemos
# matrices de 100filas constantes, y con columnas (indicado por el
# parametro "-1") que se adaptaran a la dimensión que nos deje que
# sea mayor o menor que 5
cols = lda_out_prob[lda_out_prob <= 5].reshape(100, -1)
rows = lda_out_prob[lda_out_prob > 5].reshape(100, -1)
text = ''
for i in range(100):
    # El índice -1 implica que contamos desde el final
    row = rows[i][-1]-6
    col = cols[i][-1]
    letter = MATRIX[row][col]
    text += letter

print
print 'Result for subject %d' % (subject+1)
print 'Constructed labels: %s' % text.upper()
print 'True labels      : %s' % labels
# Lo pone en mayuscula
a = np.array(list(text.upper()))
b = np.array(list(labels))
# Compara lo sacado desde el algoritmo con el feedback de los
# sujetos al hacer el experimento (el carácter que ellos realmente
# estaban pensando)
accuracy = np.count_nonzero(a == b) / len(a)
print 'Accuracy: %.1f%%' % (accuracy * 100)
acc += accuracy

print
print 'Overall accuracy: %.1f%%' % (100 * acc / 2)

### Analysis of the data

# The following part shows how to visualize interesting information of
# the data.

avgs = [None, None]
fig, axes = plt.subplots(2, 3, sharex=True, sharey=True, figsize=(9,
6))

# Itera para los datos de los dos sujetos (A y B), para hacer el
# análisis por separado y mostrarlos juntos posteriormente
for idx, file in enumerate([TRAIN_A, TRAIN_B]):
    # Primero, calcula los promedios de los datos según la clase de
    # los mismos correspondientemente con las épocas (epo) de ambos
    # sujetos. Recordamos que la variable epo tiene dos dimensiones,
    # siendo los datos segmentados para [sujeto A, sujeto B]
    avgs[idx] = proc.calculate_classwise_average(epo[idx])
    #avgs[idx] = proc.correct_for_baseline(avgs[idx], [0, 50])

# Tras hacerle los promedios según las épocas, selecciona
# unicamente tres canales centrales para su representación (después
# de haberles hecho el promedio a cada uno)

```

```

datetes = proc.select_channels(avgs[idx], ["fcz", "cz", "oz"])

for i in range(3):
    # Los puntos suspensivos en el array implica tomar de todas las
    # dimensiones
    # (...).T => traspuesta de la matriz
    axes[idx, i].plot(datetes.axes[-2], datetes.data[..., i].T)
    axes[idx, i].grid()

# Pone los titulos para cada subplot (el titulo de cada grafica)
for i in range(3):
    axes[0, i].set_title(datetes.axes[-1][i])

# Pone el eje x (ms)
axes[1, 1].set_xlabel('time [ms]')
for i in range(2):
    axes[i, 0].set_ylabel(u'voltage [a.u.]')

for i in range(2):
    axes[i, 2].yaxis.set_label_position("right")
    axes[i, 2].set_ylabel('Subject %s' % 'AB'[i])

axes[0, -1].legend(datetes.class_names)
plt.tight_layout()

# Dibujo de los sensores
def plot_scalps(epo, ivals):
    # ratio scalp to colorbar width
    scale = 10
    # Hace las epocas de los datos segun los intervalos pasados
    dat = proc.jumping_means(epo, ivals)
    n_classes = epo.data.shape[0]
    n_ivals = len(ivals)
    for class_idx in range(n_classes):
        vmax = np.abs(dat.data).max()
        vmax = round(vmax)
        vmin = -vmax
        for ival_idx in range(n_ivals):
            ax = plt.subplot2grid((n_classes, scale*n_ivals+1),
                                  (class_idx, scale*ival_idx),
                                  colspan=scale)
            plot_ax_scalp(dat.data[class_idx, ival_idx, :],
                          epo.axes[-1], vmin=vmin, vmax=vmax)
            if class_idx == 1:
                ax.text(0, -1.5, ivals[ival_idx],
                       horizontalalignment='center')
            if ival_idx == 0:
                ax.text(-1.5, 0, ['nontarget', 'target'][class_idx],
                       color='bm'[class_idx], rotation='vertical',
                       verticalalignment='center')

# colorbar (la de los voltajes)
ax = plt.subplot2grid((n_classes, scale*n_ivals+1), (0,

```

```

                                scale*n_ivals), rowspan=n_classes)
plot.ax_colorbar(vmin, vmax, label='voltage [a.u.]',
                 ticks=[vmin, 0, vmax])

for subj_idx in range(2):
    # Dibujo de la figura
    fig = plt.figure(figsize=(11, 6))
    # Los intervalos para hacer los saltos de A y B
    ivals = [JUMPING_MEANS_IVALS_A, JUMPING_MEANS_IVALS_B][subj_idx]
    # Dibujo de las cabezas
    plot_scalps(avgs[subj_idx], ivals)
    plt.tight_layout()
    fig.subplots_adjust(left=.06, bottom=.10, right=None, top=None,
                        wspace=0, hspace=0)

# Representacion de r2 en una barra según las epocas, para hacer una
# diferencia mayor entre maximos y minimos, representando todos los
# canales centrales correspondiendose los valores con los registrados
# en los dibujos previos
fig, axes = plt.subplots(2, 1, sharex=True, sharey=True)
for i in range(2):
    r2 = proc.calculate_signed_r_square(epo[i])
    # switch the sign to make the plot more consistent with the
    # timecourse. This is equivalent to reordering the classidices and
    # calculating r2
    r2 *= -1

    max = np.max(np.abs(r2))
    im = axes[i].imshow(r2.T, aspect='auto', interpolation='None',
                       vmin=-max, vmax=max)

    axes[i].set_ylabel('%s' % (epo[i].names[-1]))
    axes[i].grid()
    axes[i].set_title("Subject %s" % "AB"[i])
    cb = plt.colorbar(im, ax=axes[i])
    cb.set_label('[a.u.]')

    axes[1].yaxis.set_major_formatter(ticker.IndexFormatter(epo[i].axes[-1]))
    mask = map(lambda x: True if x.lower().endswith('z') else False,
epo[i].axes[-1])
    axes[1].yaxis.set_major_locator(ticker.FixedLocator(np.nonzero(mask)[0]))
    axes[1].xaxis.set_major_formatter(ticker.IndexFormatter(['%d' % j for
j in epo[i].axes[-2]]))
    axes[1].xaxis.set_major_locator(ticker.MultipleLocator(6))
    axes[1].set_xlabel('%s [%s]' % (epo[i].names[-2], epo[i].units[-2]))

plt.tight_layout()
plt.show()

```